bpost shipping manager

Integration Manual

Version 2.0

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

Table of Contents

1 Document modifications

The following modifications have been introduced into the application:

Document version	Change
2.0	Callback URL: It is now possible to send the confirm, error and cancel URL as parameters when requesting to show the iFrame. This will allow efficient multi-shop integrations and/or usage of different domains during development/testing.
2.0	Company name: It is possible to send through a company name when requesting to show the iFrame. This field will be placed on the label REMARK: the webservice is not yet updated to include this feature.
2.0	Returned parameters: More parameters are returned when the iFrame is closed. They include notification information and selected options/services

2 Glossary of Terms

Throughout this document, the following terms and abbreviations are used. They are explained in this table.

Term	Definition
WS	Web service: Describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone.
ERP	Enterprise Resource Planning: The technology solutions for large companies or other organizations to manage their resources, including customers, supplies, accounting, and other "back office" activities.
LCI (IN/OUT)	Large Customer Interface: Information used by bpost to generate labels.
Lightbox	A simple, unobtrusive (java)script used to overlay websites. It is easy to setup and works on all modern browsers.
PUGO	Pick-up & Go: bpost delivery method. For more information please visit www.bpost.be. This name is no longer valid and is replaced by "Bpack@bpost"
Bpack 24/7	bpost delivery method. For more information please visit www.bpost.be.
API	Application Programming Interface: a particular set of rules and specifications that software programs can follow to communicate with each other.
REST	Representational State Transfer: an approach for getting information content from a Web site by reading a designated Web page that contains an XML file that describes and includes the desired content.
URI	Uniform Resource Identifier: a string of characters used to identify a name or a resource on the Internet.
XML	eXtensible Markup Language
XSD	XML Schema Definition
BPACK@bpost	The new name of PUGO, a delivery method of bpost

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

3 Introduction

This document describes the integration procedure of the bpack Shipping Manager.

Please note that the shipping manager Integration Manual does not explain how to use the Back End of the shipping manager. Please read the *bpack Shipping Manager user guide* if you want to work with the Back End interface.

3.1 Required knowledge

In order to use this manual you need knowledge of the following topics:

- XML
- Web services
- HTTPS

XML stands for eXtensible Markup Language. For an introduction, go the W3 Schools XML Tutorial at http://w3schools.com/xml/default.asp.

A Web service is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards." For an introduction, go to the W3 Schools Web Services Tutorial at http://w3schools.com/webservices/default.asp.

REMARK
The bpack Shipping Manager runs in an HTTPS environment, in order to safeguard the communication of private data. bpost strongly recommends using https and does not support integrations in http environments.

3.2 bpack Shipping Manager solution

The bpack Shipping Manager is a bpost solution, available for national and international shipping. It allows your business to manage shipping and returns and to print labels. The bpack Shipping Manager solution contains a Front End and a Back End solution.

3.2.1 Front End

The bpack Shipping Manager Front End is an online solution for your e-business to offer bpost delivery options to consumers.

Figure 1 bpack Shipping Manager Front End solution

The Front End fits seamlessly into the order process, which is shown in Figure 1. The following steps show how the process works:

First the consumer selects his desired product(s) in the web shop. Next the consumer enters his credentials and performs a check out.

During the checkout process, when selecting the delivery method / address, the consumer is redirected via URL (POST parameters) to the shipping manager Front End. In the Front End he selects the appropriate delivery method and options. All the information is automatically sent to the bpost systems and all the information is also returned via URL redirect (POST parameters) to the web shop.

The consumer then comes back to the web shop to validate and pays the order. Afterwards, the web shop can send a confirmation to bpost in order to validate the sale after payment via another POST URL redirection or via a REST web service.

3.2.2 Back End

The bpack Shipping Manager Back End is a web-based solution that allows your e-business to:

- Manage shipments
- Select additional options
- Generate labels
- Link web orders to parcel shipping
- Access Track & Trace information
- Manage returns

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

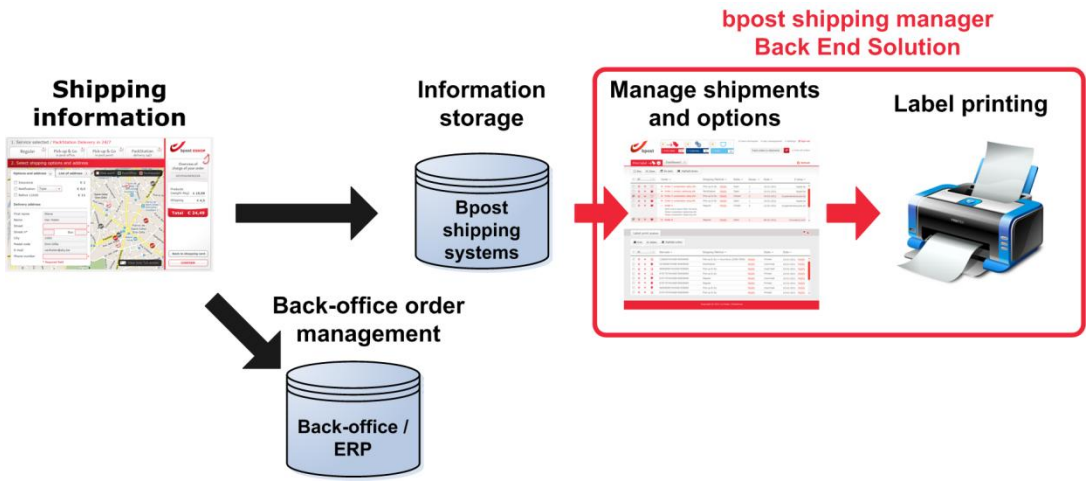


Figure 2 bpack Shipping Manager Back End solution

In Figure 2 we show you the process in which the Back End is placed. The following steps show how the process works:

The shipping information, which is returned either by an integrated Front End or by REST web services, can be stored in your Back-office or ERP system and is automatically stored in the systems of bpost. This covers information such as: shipping address, product, client name, etc.

It is possible to manage the information through the shipping manager Back End which is hosted by bpost. E-tailers can access it via the bpost portal to manage orders, select options and generate labels. Generated labels can be printed by the shipping manager and can be used to launch the logistic flow. It is possible to follow the shipped parcels with the integrated Track & Trace tool.

Apart from this solution, the bpack Shipping Manager allows the deep integration of its functionalities into ERP or other back-office systems. This deep integration is thoroughly explained throughout the following sections and in the chapter *Web Services*.

3.3 Four integration solutions

bpost offers an easy integration and a deep integration solution for both the Front End and the Back End parts of the shipping manager, which makes a total of four integration solutions. In Figure 3 an overview of the four integration solutions is provided.

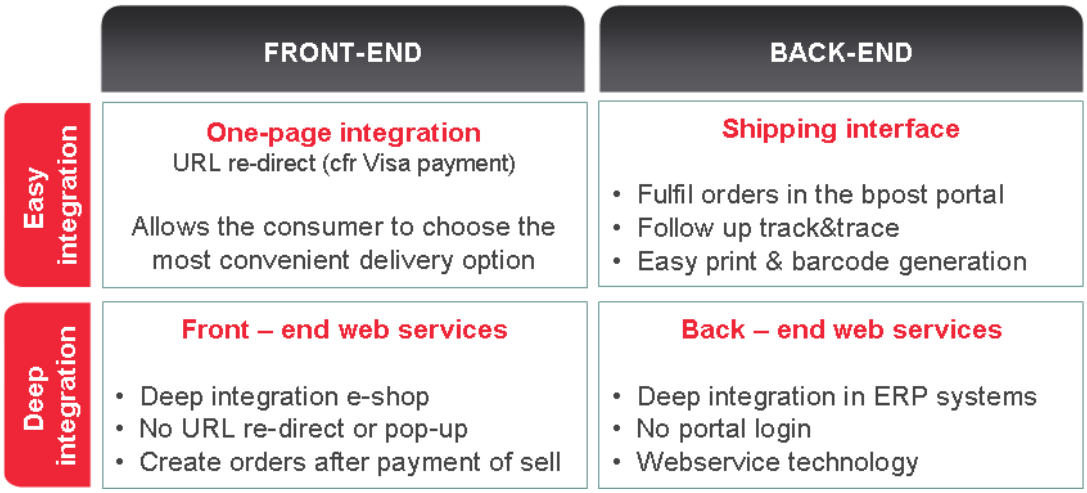


Figure 3 Four integration solutions

3.3.1 Front End easy integration

The easy integration of the Front End is implemented via a URL re-direct similar to popular payment services. On this re-direct page the consumer can choose the most convenient delivery option.

Go to the chapter *Front End Easy Integration* to read the details of how to implement this integration solution.

3.3.2 Back End easy integration

The easy integration of the Back End is accessible as a web-based application via the bpost portal.

The shipping manager Back End allows e-tailers to manage shipping and returns, generate and print labels and access Track & Trace information.
Since the Back End easy integration only involves a login to the bpost portal in order to access the Shipping interface, there are no further details to be discussed concerning this integration solution. Please read the *bpack Shipping Manager user guide* if you want to use the easy integration of the Back End.

3.3.3 Front End deep integration

The deep integration of the Front End is implemented via REST web services that provide real-time data transfer between the web shop and bpost.

This deep integration of the Front End into your e-shop doesn't use a URL re-direct or pop-up, but makes the same technology as the easy integration available as web services.

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

Go to the chapter *Web Services* to read the details of how to use the web services to implement the deep integration of the Front End.

3.3.4 Back End deep integration

The deep integration of the Back End is implemented as web services that provide real-time data transfer between the e-tailer back-end systems and bpost.

The web services of the deep integration allow e-tailers to integrate all shipping and labelling management in their ERP systems or back-office applications.

It is not required to login into the shipping interface hosted on the bpost portal. The same technology as the easy integration is made available as web services, which means it is 100% transparent.

Go to the chapter *Web Services* to read the details of how to use the web services to implement the deep integration of the Back End.

3.4 Four integration scenarios

With the four integration solutions, i.e. easy and deep integration of Font End and Back End, bpost offers four combinations of the solutions as possible scenarios you can deploy in your e-business. Below we list the features and characteristics of each scenario.

Scenario 1: easy Front End and easy Back End

- Show different and up to date delivery methods in the checkout process via the bpost frame, perfectly integrated into the look & feel of the webshop
- Information automatically stored in the bpost Back End
- Fulfil orders and print labels via the bpost Shipping interface
- Optional: Confirmation call after payment via Web Service

Scenario 2: deep Front End and easy Back End

- Only use one delivery method (bpack@home) and create orders directly after payment by consumer via web services
- Store information in the bpost Back End
- Fulfil orders and print labels in the bpost Shipping interface

Scenario 3: deep Front End and deep Back End

- Integrate directly in your ERP system and/or during check-out (after payment)
- Use bpost web services to send and request information
- Store labels generated by bpost (and auto-print) in your own ERP system
- No need to use bpost applications for the logistic process

Scenario 4: easy Front End and deep Back End

- Show different and up to date delivery methods in the checkout process via the bpost frame, perfectly integrated into the look & feel of the webshop
- Use ERP integration for back-end via web services
- Use bpost web services to send and request information
- Store labels generated by bpost (and auto-print) via own ERP system

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

We will now have a more in-depth look at the scenarios.

3.4.1 Scenario 1 – story line

Below are the steps describing the process of scenario 1 the easy Front End and easy Back End integration.

A1 Setup
After signing a contract, the customer integrates the shipping application Front End in a web shop.

A2 Configure
The customer configures the admin panel of the shipping application according to the needs of the organization.

A3 Activate
The customer is activated in the shipping manager.

B1. Request to show
The web shop sends out a secured request to show to front end UI.

B2. Validate
The request is validated based on account ID, security key and configuration.

B3. Show front end
The front end is shown to the consumer on the web shop.

B4. Select shipment
The consumer selects the most convenient shipment method.

B5. Save shipping data
All relevant shipping information is automatically saved in the bpost Back End. Some information is returned to the web shop.

B6. Confirm Order (WS / URL redirection)
Optional: The web shop confirms the order after payment step via web service.

C1. Finish shipping data
The customer logs into the Back End within the portal and adapts the shipping data. Labels, Track & Trace, etc. can be managed.

C2. Create labels
Labels are automatically generated by the bpost systems.

3.4.2 Scenario 2 – story line

Below are the steps describing the process of scenario 2 the deep Front End and easy Back End integration.

A1. Setup
After signing a contract, the customer integrates the shipping application in a web shop.

A2. Configure
The customer configures the admin panel of the shipping application according to the needs of the organization.

A3 Activate
The customer is activated in the shipping manager.

B1. Create Order (WS)
The web shop sends information to bpost via a secured web service.
B2. Validate
The request is validated based on account ID, security key and configuration.
B3. Save shipping data
All relevant shipping information is saved in the bpost Back End. Some information is returned to the web shop.

C1. Finish shipping data
The customer logs into the Back End within the portal and adapts the shipping data. Labels, Track & Trace, etc. can be managed.
C2. Create labels
Labels are automatically generated by the bpost systems.

3.4.3 Scenario 3 – story line

Below are the steps describing the process of scenario 3 the deep Front End and deep Back End integration.

A1. Setup
After signing a contract, the customer integrates the shipping application in a web shop.
A2. Configure
The customer configures the admin panel of the shipping application according to the needs of the organization.
A3 Activate
The customer is activated in the shipping manager.

B1. Create Order (WS)
Optional: The web shop sends information to bpost via a secured web service. This information can be sent from the ERP or from the web shop.
B2. Validate
The request is validated based on account ID, security key and configuration.

C1. Create Labels (WS)
All relevant information to create a box is sent to the bpost systems via a secured web service. The e-tailer is able to ask a set of data (labels, barcode, etc.).
C2. Send label
If requested, automatic generated labels are returned to the e-tailer instantly. They can be saved or directly printed.

3.4.4 Scenario 4 – story line

Below are the steps describing the process of scenario 4 the easy Front End and deep Back End integration.

A1 Setup

After signing a contract, the customer integrates the shipping application in a web shop.

A2 Configure

The customer configures the admin panel of the shipping application according to the needs of the organization.

A3 Activate

The customer is activated in the shipping manager.

B1. Request to show

The web shop sends out a secured request to show to front end UI.

B2. Validate

The request is validated based on account ID, security key and configuration.

B3. Show front end

The front end is shown to the consumer on the web shop.

B4. Select shipment

The consumer selects the most convenient shipment method.

B5. Save shipping data

All relevant shipping information is automatically saved in the bpost Back End. Some information is returned to the web shop.

B6. Confirm Order (WS / URL redirection)

Optional: The web shop confirms the order after payment step via web service.

C1. Create Labels (WS)

All relevant information to create a box is send to the bpost systems via a secured web service. The e-tailer is able to ask a set of data (labels, barcode, etc.).

C2. Send label

If requested, automatic generated labels are returned to the e-tailer instantly. They can be saved or directly printed.

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

3.5 ICT Impact

Below we provide an overview of the impact of each possible integration scenario on an ICT level:

Impact of scenario 1: easy Front End and easy Back End

- Integration of iFrame in website (e.g. via lightbox pop-up)
- Send shipping parameters via Front End (URL redirection – POST parameters)
- Receive shipment information via return URL (POST parameters)
- Optional: Confirm order after payment via web service or URL redirection (POST)
- Labels generated within Back End
- LCI-IN file automatically generated within Back End
- Track and Trace available in the Back End

Impact of scenario 2: deep Front End and easy Back End

- Send shipping parameters via web service [Create Order (WS)]
- Optional: Confirm order after payment
- Labels generated within Back End
- LCI-IN file automatically generated within Back End
- Track and Trace available in the Back End

Impact of scenario 3: deep Front End and deep Back End

- Send parameters via web service during checkout in web shop [Create Order (WS)]
- Send shipping parameters from ERP via web service and request barcode + label [Create Labels (WS)]
- LCI-IN and label automatically generated by system
- Track & Trace available upon web service request [Tracking (WS)]
- Update status of shipment via web service [Modify Order Status(WS)]

Impact of scenario 4: easy Front End and deep Back End

- Integration of iFrame in website (e.g. via lightbox pop-up)
- Send shipping parameters via Front End (URL redirection – POST parameters)
- Receive shipment information via return URL (POST parameters)
- Optional: Confirm order after payment via web service or URL redirection (POST)
- Send shipping parameters from ERP via web service and request barcode + label [Create Labels (WS)]
- LCI-IN and label automatically generated by system
- Track & Trace information available upon web service request [Tracking (WS)]
- Update status of shipment via web service [Modify Order Status(WS)]

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

4 General configuration / Setup

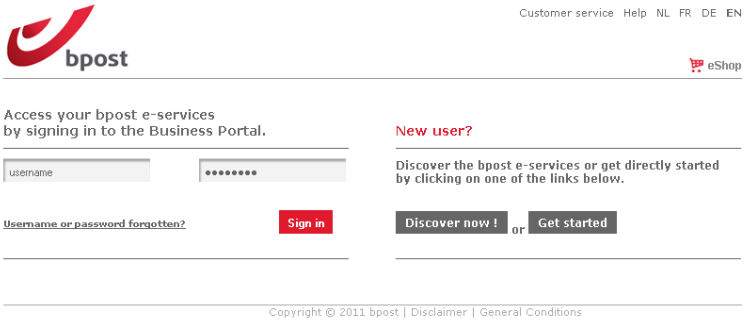
This chapter describes the general setup / configuration that is required to use the bpack Shipping Manager. This setup has to be followed for all different types of integrations.

Before using the bpack Shipping Manager it is required to set up your account in the Back End by logging into the bpost portal and configuring the settings in the Back End Shipping interface. For instructions on how to use the Shipping interface please read the *bpack Shipping Manager user guide*.

4.1 Logging in

Open an internet browser and go to the bpost portal at the following URL:

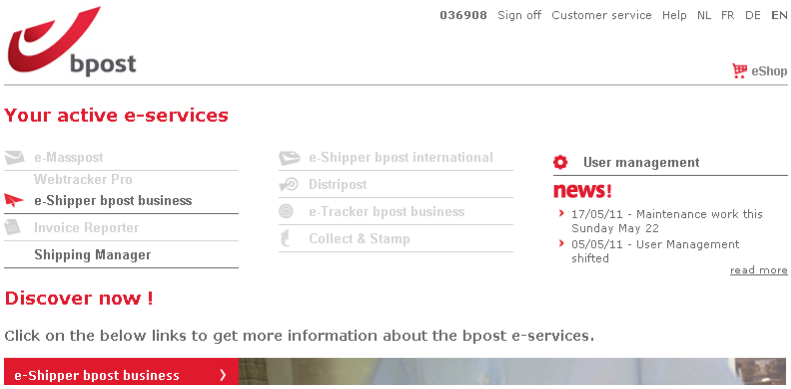
https://www.bpost.be/portal/login



Choose the language, enter your Account ID and password and click the **Sign in** button to login.

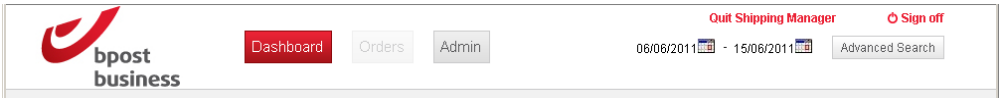
4.2 Configuration

After logging in, all active e-services on the bpost portal are shown.

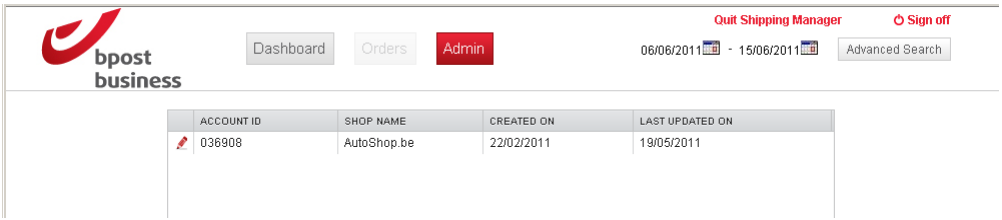


Click **shipping manager** to open the Back End Shipping interface.

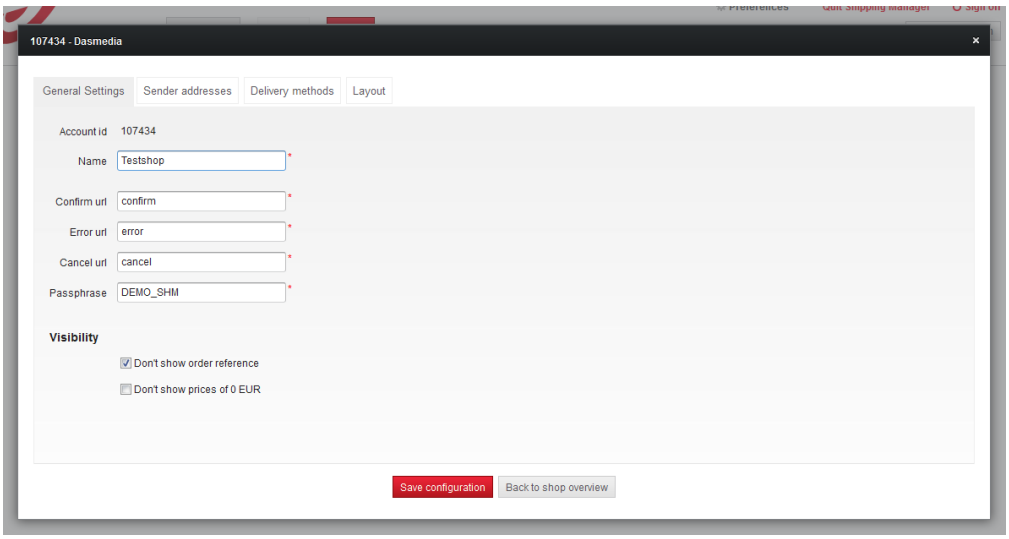
bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1



Click the **Admin** button.



Click the pencil icon in front of your Account ID.



[1] In the **General Settings** tab make sure the following fields are filled in:

Confirm URL: in this field, key in the URL of the confirmation web page on the web shop. This is the web page that will be shown to the consumer, after he has clicked the **Confirm** button in the Front End pop-up. This page will receive all HTTP POST parameters described in the next section.

Error URL: in this field, enter the URL of the error web page on the web shop. This screen is shown when an error occurs.

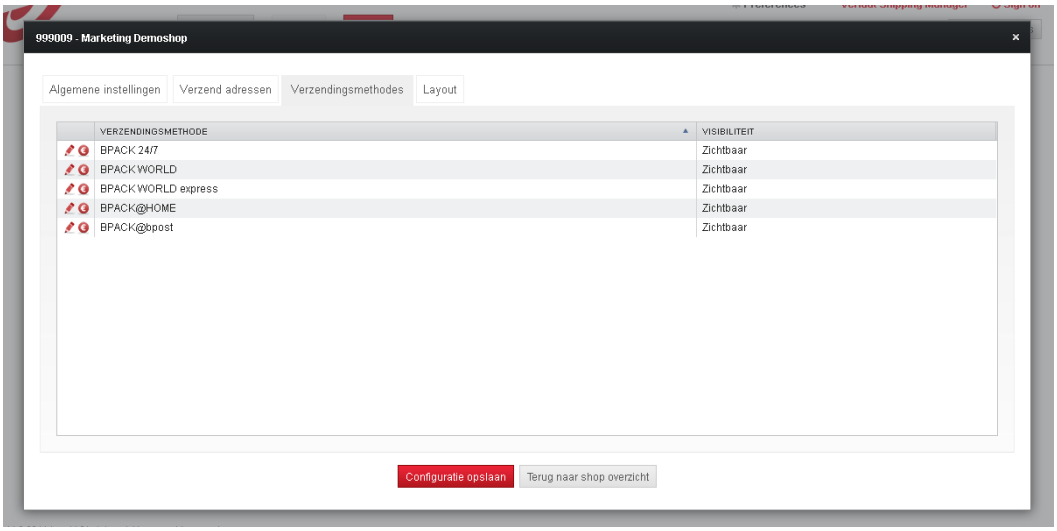
Cancel URL: in this field, enter the URL of the cancel action that the web shop provides. It will be shown when the consumer clicks the **Back to shopping** button on the Front End pop-up.

Passphrase: in this field, key in the password that will be used for generating the security hash. Please modify the default value of this field and by

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

preference, use a password that contains a combination of letters and numbers and has at least 10 characters in length.

[2] Click the **Delivery Methods** tab.

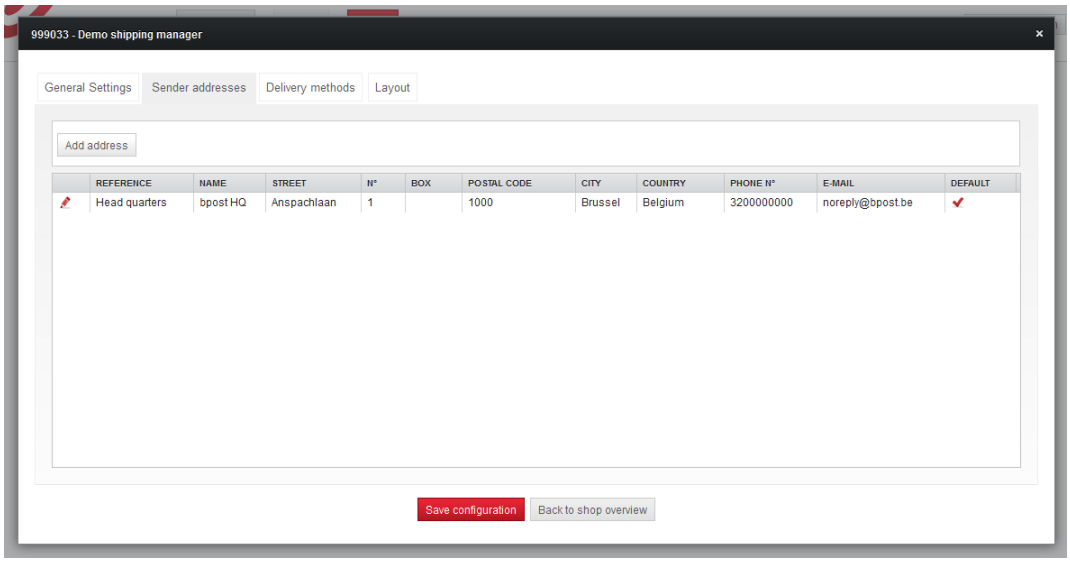


Make sure you have configured your delivery methods. Set the delivery methods you want to show to **Visible**.

For more information on the different delivery methods and their usage, please refer to our *bpack Shipping Manager user manual*.

[3] Click the **Sender Address** tab

The current default value must contain an existing Belgian address. If not correctly configured, the sent goods will not be returned in case of non delivery and our messaging services will not send messages when the goods are delivered in the bpack@bpost location.



bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

5 Front end integration

This chapter describes how to setup the front end integration of the bpack Shipping Manager, allowing the integration of the bpack delivery methods into your checkout process.

5.1 Parameters

When you have configured your account and your delivery methods, you are able to use the Front End redirect page. You need to perform the following steps to successfully call the Front End application:

1. Provide the necessary HTTP POST parameters
2. Calculate a checksum
3. Send the parameters and the checksum via HTTP POST

5.1.1 HTTP POST parameters

The web shop needs to send the parameters specific to your account and the order to the following URL:

https://shippingmanager.bpost.be/ShmFrontEnd/start

The following fields are mandatory and their values are filled if available. All values are case sensitive:

- accountId
- action (value = START)
- orderReference
- customerCountry

The value of the **action** field needs to be **START** to open the redirect page of the shipping manager.

The table below lists all the possible hidden parameters you can send to the redirect page of the Front End. All values listed are case sensitive.

Name	Required	Hashed	Allowed Values	Description
accountId	**Required**	Yes		Your unique customer ID of bpost P&E
action	**Required**	Yes	START CONFIRM	START opens the redirect page CONFIRM confirms the order
lang	Optional	No	NL FR EN DE Default: language settings of your browser	Language NL = Dutch FR = French EN = English DE = German

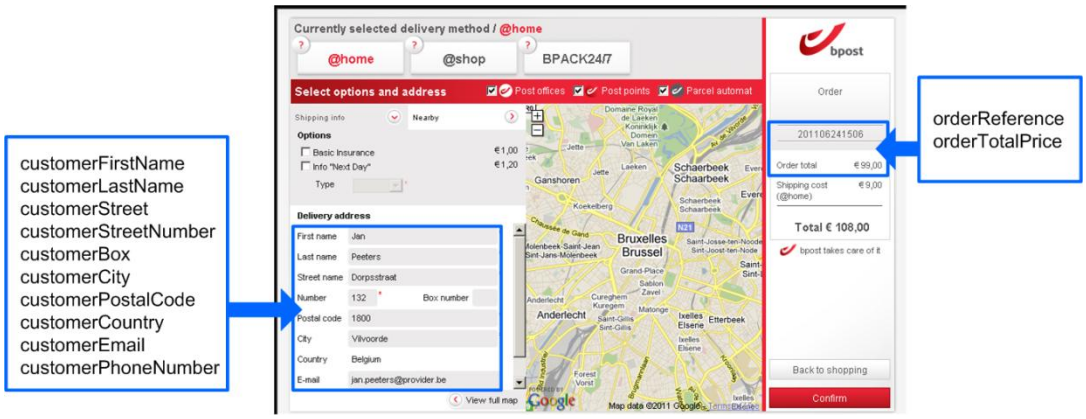
bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

checksum	**Required**	No	SHA-256 hash	Checksum of fields. **See 5.1.2 Checksum**
orderReference	**Required**	Yes	Max length = 50	Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
costCenter	Optional	Yes	Max length = 50	This information is used on your invoice and allows you to attribute different cost centers (e.g. different shops, warehouses, suppliers, …)
orderTotalPrice	Optional	No	Euro cents	Total price of the basket order in Euro cents (excluding shipping costs)
customerFirstName	Optional	No		First name of the customer
customerLastName	Optional	No		Last name of the customer **The first name and the last name together should not be longer than 40 characters. Only 40 characters are shown on the label**
customerCompany	Optional	No	Max length =	Name of the company of the receiver
customerStreet	Optional	No	Max length = 40	Street name of the customer
customerStreetNumber	Optional	No	Max length = 8	Street number of the customer
customerBox	Optional	No	Max length = 8	Box number of the customer
customerCity	Optional	No	Max length = 40	City of the customer
customerPostalCode	Optional	No	Max length = 32	Postal code of the customer
customerCountry	**Required**	Yes	Uppercase 2 character country ISO code	Country of the customer
customerEmail	Optional	No	Max length = 50	E-mail address of the customer
customerPhoneNumber	Optional	No	Max length = 20	Mobile phone number of the customer
orderLine	Optional	No		The items that are included in the order. Order lines are shown in

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

				the back end of the shipping manager and facilitate the use of the tool. Multiple orderLines are available for a single order.
orderWeight	Optional	Yes	Integer	Weight of the order in grams
deliveryMethodOverrides	Optional	Yes		Overrides of delivery method for specific order
extra	Optional	No		Additional parameters related to your web shop. Information not used by bpost and returned in the confirm/error/cancel URL.
extraSecure	Optional	Yes		Additional parameters related to your web shop included in the hash. Same as the extra parameters, but hashed.
confirmUrl	Optional	No		URL to where the customer is redirected once the iFrame is closed. If empty, the default value is used
cancelUrl	Optional	No		URL to where the customer is redirected if the button "back to shopping car" is clicked. If empty, the default value is used
errorUrl	Optional	No		URL to where the customer is redirected if an error occurs. If empty, the default value is used

The following screenshot shows where the parameters are shown on the Front End pop-up:



5.1.2 Checksum

The checksum is a 256 bit Secure Hash Algorithm (SHA-256) in UTF-8 encoding of the following required fields: accountId + action + customerCountry + orderReference + passphrase. All the fields need to be passed in alphabetical order.

In case one or more of the following optional fields is used in the form, they should also be included in the checksum calculation:

- costCenter
- deliveryMethodsOverrides
- extraSecure
- orderWeight

These optional fields need to be inserted into the sequence alphabetically.

For example: if the accountId is 123456, the costCenter is Online Shop, the customerCountry is BE, the orderReference is 201106161621 and the password is MyPassPhrase, you need to generate the checksum of the following string of the fields concatenated by the ampersand (&):

```
accountId=123456&action=START&costCenter=Online
Shop&customerCountry=BE&orderReference=201106161621&MyPassPhrase
```

This string will give you the following checksum hash:

```
f9b5e5ab0fbe6fce786ccbff4ceb763bc125152fe532283dca7a93f4875ef44f
```

For more information on how to generate SHA256, please refer to your favourite search engine.

REMARK: As shown in the above example, the passphrase is added to the end of the checksum calculation string by placing an ampersand, followed by the passphrase itself. It is not required to put "&passphrase=MyPassPhrase".

5.1.3 Order lines

The format of the orderLine parameter consists of the order line description and the number of items piped (|):

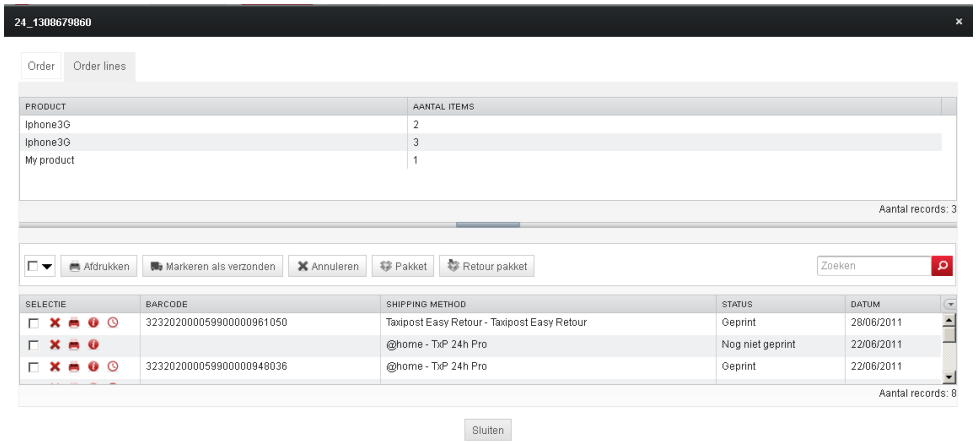
orderLineDescription|numberOfItems

- orderLineDescription is a short description. This description could be used by the person picking the order, to identify the items. If available, add the item code in front of the description, to facilitate the order management.
- numberOfItems is an integer and represents the number of items.

Example: orderLine=08815-GSM-iPhone-3GS|2

The example means that the consumer ordered 2 iPhones of type 3GS, product category GSM, with item number 08815.

The orderLine parameter can be sent multiple times. Every variable will be listed as a new order line in the shipping manager Back End (see next figure).



bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

5.1.4 Delivery method overrides

You can use a number of fields of the type deliveryMethodOverrides containing the delivery method name, the visibility and the price in Euro cent. The following example shows the accepted format of the field and a few examples of values:

deliveryMethodOverrides (1): Accepted format: name|visibility|[priceInEuroCent]
deliveryMethodOverrides (2): Example: Regular|VISIBLE|5400
deliveryMethodOverrides (3): Example: Regular|INVISIBLE
deliveryMethodOverrides (4): Example: Regular|GREYED_OUT

The table below lists all possible delivery method names you can use as a value in the deliveryMethodOverrides fields and their corresponding labels as bpost delivery services.

Delivery Method Name	Label	Example
bpack EXPRESS	bpack World Express	bpack EXPRESS|VISIBLE|10000
bpack BUSINESS	bpack World	bpack BUSINESS|INVISIBLE
Parcels depot	bpack 24/7	Parcels depot|VISIBLE|4500
Pugo	bpack@bpost	Pugo|GREYED_OUT
Regular	bpack@home	Regular|VISIBLE|5400

The deliveryMehodsOverrides parameter can be sent multiple times. Every variable will be interpreted separately.

ATTENTION The deliveryMethodOverrides fields must be included in the checksum hash. See the table in section *5.1.1 HTTP POST parameters* and the explanation of the checksum generation below in section *5.1.2 Checksum*.
The parameters need to be send in alphabetical order within the checksum in order to prevent a 2000 error.

5.2 Sending parameters via HTTP POST

The link between your web shop and the shipping manager Front End has to be established on the page where you offer the consumer the possibility to choose the delivery method. Into that HTML page you need to integrate a form with hidden HTML fields containing the parameters listed above. Send the parameters and the checksum to the Front End at the following URL:

https://shippingmanager.bpost.be/ShmFrontEnd/start

Use an HTML form with a POST method and the URL in the action attribute. The following example shows you the HTML code to achieve this:

```
<form id="myForm" method="POST" target="shmFrame"
action="https://shippingmanager.bpost.be/ShmFrontEnd/start">
</form>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

A parameter that you want to send to the Front End needs to be put in a hidden input element between the <form></form> tags. The following example shows the accountId parameter in a hidden input element:

```
<form id="myForm" method="POST" target="shmFrame"
action="https://shippingmanager.bpost.be/ShmFrontEnd/start">
<input type="hidden" name="accountId" value="123456"/>
</form>
```

All the parameters and the checksum need to be placed in separate hidden input elements in the HTML form. The HTML code below contains an example of how you can implement the bpost logo, the form and the parameters in your web shop to call and open the Front End.

```
<p>We currently deliver our product with bpost. By clicking on the icon below you will
be able to define your preferred delivery method.</p>
<form id="myForm" method="POST" target="shmFrame"
action="https://shippingmanager.bpost.be/ShmFrontEnd/start">
<input type="image" id="logobpost"
src="http://www.bpost.be/site/nl/residential/parcels/pickup/handle-with-
care_2groot.jpg" style="cursor:pointer" alt="Submit button"/>
<input type="hidden" name="lang" value="EN"/>
<input type="hidden" name="accountId" value="999033"/>
<input type="hidden" name="action" value="START"/>
<input type="hidden" name="orderReference" value="12345 Test Order"/>
<input type="hidden" name="orderTotalPrice" value="66490"/>
<input type="hidden" name="customerFirstName" value="Jan"/>
<input type="hidden" name="customerLastName" value="Peeters"/>
<input type="hidden" name="customerStreet" value="Dorpsstraat"/>
<input type="hidden" name="customerStreetNumber" value="132"/>
<input type="hidden" name="orderLine" value=" Iphone 4 16GB Black|1"/>
<input type="hidden" name="orderLine" value="Earphones|1"/>
<input type="hidden" name="customerBox" value=""/>
<input type="hidden" name="customerCity" value="Vilvoorde"/>
<input type="hidden" name="customerPostalCode" value="1800"/>
<input type="hidden" name="customerEmail" value="jan.peeters@provider.be"/>
<input type="hidden" name="customerPhoneNumber" value="0032499123456"/>
<input type="hidden" name="customerCountry" value="BE"/>
<input type="hidden" name="checksum"
value="61ddef9ff679508d455a5f9f451c3a2bfe8b721f2812ddb0b1990325a2039bb5"/>
</form>
```

5.3 Showing the Front End pop-up

In order to show the Front End as a pop-up on top of your web shop page you need to include code to create an iframe.

Create an iframe that has the same name as the target of your HTML form. The HTML code below show an iframe with the name "shmFrame", which is the same as the target we used in the HTML code for the form above.

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
<iframe id="shmFrame" name="shmFrame" width="790" height="520"
style="border:none;"></iframe>
```

The code example below shows one method to implement the iframe in a pop-up. By using CSS, it is also possible to modify the div attributes, allowing for instance a "lightbox" pop-up integration. (The script example only works when the jquery plugin is installed). Another solution is to unfold the iFrame when a radio button is selected, with the iFrame completely integrated (so no pop-up).

```
<div id="showme">
<iframe id="shmIFrame" name="shmFrame" width="0" height="0"></iframe>
</div>
<script>
$("#logobpost").click(function() {
$("#myForm").submit();
$("#shmIFrame").attr("width","800");
$("#shmIFrame").attr("height","600");
$("#shmIFrame").css("position","absolute");
$("#shmIFrame").css("z-index","500");
$("#shmIFrame").css("width","800px");
$("#shmIFrame").css("left","50%");
$("#shmIFrame").css("margin-left","-400px");
$("#shmIFrame").css("top","50px");
});
</script>
```

A sample integration using php, javascript and jquery is available upon request. Please contact shippingmanager@bpost.be if you are interested in receiving this information.

5.4 Returned parameters

The following table lists the parameters that are returned by the shipping manager to the web shop when the consumer clicks the **Confirm** button. The address information that will be returned in the parameters is the chosen delivery address.

Name	Allowed Values	Description
orderReference	Max length = 50	Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
costCenter	Max length = 50	This information is used on your invoice and allows you to attribute different cost centers (e.g. different shops, warehouses, suppliers, …)
orderTotalPrice	Euro cents	Total price of the basket order in Euro cents (excluding shipping costs)
customerFirstName		First name of the customer

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

customerLastName		Last name of the customer **The first name and the last name together should not be longer than 40 characters. Only 40 characters are shown on the label**
customerMemberId		Bpack24/7 member identifier
customerStreet	Max length = 40	Street name of the customer
customerStreetNumber	Max length = 8	Street number of the customer
customerBox	Max length = 8	Box number of the customer
customerCity	Max length = 40	City of the customer
customerPostalCode	Integer Max length = 32	Postal code of the customer
customerCountry	Uppercase 2 character country code	Country of the customer
customerEmail	Max length = 50	E-mail address of the customer
customerPhoneNumber	Max length = 20	Mobile phone number of the customer
customerPostalLocation		Name of Post Point or Bpack 24/7 automate in case delivery method "BPACK@bpost" of "Bpack 24/7" is selected
customerRcCode		RC code to identify a Post Point or Bpack 24/7 automate
orderLine		The items that are included in the order. Order lines are shown in the back end of the shipping manager and facilitate the use of the tool. Multiple orderLines are available for a single order.
orderWeight	Integer	Weight of the order in grams
extra		Additional parameters related to your web shop. Information not used by bpost and returned in the confirm/error/cancel URL.
extraSecure		Additional parameters related to your web shop included in the hash. Same as the extra parameters, but hashed.
deliveryMethod		Delivery method selected by the consumer
deliveryMethodPriceDefault		Default price of the selected delivery method
deliveryMethodPriceOverride		Price override of the selected delivery method
deliveryMethodPriceTotal		Total price of the selected delivery method
Selected services (if applicable for your integration)		**Please refer to 8 APPENDIX - Returned service information**

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

5.5 Confirming the order

It is recommended to confirm the order to bpost. This will change the status of the order in the back-end system, allowing you to identify which orders have been paid and which remain pending. bpost is not able to retrieve what happens between the selection of the delivery method and the actual payment of the order. If the confirmation is not provided, there is no difference between orders that were cancelled between the delivery step and the payment step, which may lead to wrong deliveries.

To confirm the order, a new http request should be performed, with "CONFIRM" as value in the action field. Please refer to 5.1.1 HTTP POST parameters for more information on how to send the parameters.

It is also possible to confirm or cancel an order by using web services. Please refer to chapter 6.4 Modify Order Status Web Service.

5.6 Backup solution

bpost recommends clients to create a backup solution in order to be able to offer a delivery method to their customers in case the shipping manager fails to respond. If the web shop executes an HTTP request, but the shipping manager doesn't respond, it is recommended that the web shop show a standard form with fields in which the customer can enter the delivery address.

There will be no loss of information and it is possible to manually insert the information into the bpost systems.

5.7 Error codes

When the request sent to open the redirect page is invalid, an error code will be shown. The table below lists the error codes.

Error Code	Refers to	Type	Description
1110	orderReference	Required	Order reference is required but it hasn't been supplied
1140	customerCountry	Required	Customer country code is required but it hasn't been supplied
1210	orderReference	Format error	Order reference is too long
1211	orderTotalPrice	Format error	Order total prince contains invalid integer (must be in eurocent)
1212	costCenter	Format error	Order cost center is too long
1220	orderLine	Format error	Order line has invalid format
1221	orderLine	Format error	Order line contains invalid price (must be in eurocent)
1222	orderWeight	Format error	Order weight needs to be sent in grams and should not contain decimal limiter (, or .)
1230	deliveryMethodOverrides	Format error	Delivery method overrides has invalid format
1231	deliveryMethodOverrides	Format error	Delivery method overrides has invalid visibility

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

1232	deliveryMethodOverrides	Format error	Delivery method overrides has invalid price (must be eurocent)
1240	customerCountry	Format error	Customer country code is invalid
1251	customerLastName	Format error	Last name is too long
1252	customerStreet	Format error	Street name is too long
1253	customerStreetNumber	Format error	Street number is too long
1254	customerBox	Format error	Box number is too long
1255	customerCity	Format error	City name is too long
1256	customerPostalCode	Format error	Postal code is too long
1257	customerPhoneNumber	Format error	Invalid phone number
1258	customerEmail	Format error	Invalid email
1310	orderReference	Data error	Order reference not found
1311	N/A	Data error	Order is not in the correct state (must be Pending)
1320	customerCountry	Data error	Customer country code is not correctly configured in back end
2000	N/A	Security error	Access denied (authentication failed) This can be due to a miscalculation of the checksum + value of the required fields, a wrong identifier or a wrong password (common error: usage of bpost portal login instead of Shipping Manager passphrase)

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

6 Web Services

In this chapter we discuss the Shipping API that is used to implement the deep integration of the Front End and the Back End. This document describes version 2 of the web services. For older versions of the web services, please read previous versions of this document.

6.1 Shipping API implementation

6.1.1 REST

REpresentational State Transfer (REST) software architecture style is used to expose shipping manager resources as services to the external parties of bpost.

6.1.2 Protocol

Although REST is an architectural style which is not bound to a particular technology, in practice the HTTP architecture is used. Web Services offered by the shipping manager are then implemented by sending and/or receiving XML documents over the HTTP(s) Protocol.

Resources (business entities, such as the order) are addressed by a Uniform Resource Identifier (URI). These resources can then be manipulated with the standard HTTP operations POST, GET, PUT and DELETE. These requests will map to standard CRUD operations as illustrated in the table below:

CRUD	HTTP	Action
Create	**POST**	Create a sub-resource "under" the given URI. The resource representation is passed in the request and the address (URI) of the newly created resource is returned in the response.
Read	**GET**	Retrieve the current state of the resource at the given URI. The resource representation is returned in the response.
Update	**PUT**	Initialize or update the "state" of a resource at the given URI. The complete resource representation is passed in the HTTP request.
Delete	**DELETE**	Delete a resource at a given URI. Afterwards the URI is no longer valid.

6.1.3 Endpoint

Web Services of the shipping manager application are accessible at the following URL:

https://api.bpost.be/services/shm

6.1.4 Versioning

The versioning of a web service operation is achieved by sending requests and accepting responses having a specific media-type defined by the following examples:

application/vnd.bpost.<servicefamily>-<version>+<format>
application/vnd.bpost.shmOrderCreation-v2+XML
application/vnd.bpost.shmOrderChange-v1+XML

Where the version identifier is a "v" followed by a whole number. We only distinguish between major versions. Minor versions have to be backwards compatible or else they are a major version by definition. The first version is v1, the next v2, the next v3 and so on.

This media-type value must then be set accordingly on the **Accept:** and **Content-Type:** headers of the HTTP operation.

6.1.5 XML Validation

The structure of the XML request and response messages must be validated against a schema definition. The XSD file used to describe the various XML elements exchanged between the external parties and bpost can be found at the end of this document.

One XSD file contains the definition of one version of the data to be sent in requests and responses. Eventually, this XSD will import other XSDs for bpost common definitions like addresses, names … Each version of the data has its own namespace.

Elements defined in the http://schema.post.be/api/shm/v2/ define the API version 2 of the SHM application.

In case the request contains an XML and that XML is valid against the XSD file defined in chapter 7 shm-deep-integration-v2.xsd of this document, the service is executed and a response will be sent back to the caller.

In case the request contains an invalid XML, the following response should be expected by the caller of the service:

```
HTTP/1.1 400 Bad Request
Server: Apache-Coyote/1.1
Content-Type: application/vnd.bpost.shmXMLSchemaValidationException-v1+XML
Content-Length: 461
Date: Wed, 27 Apr 2011 14:28:24 GMT
Connection: close

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:xsdValidationException xmlns="http://schema.post.be/common/exception/v1/"
xmlns:ns2="http://schema.post.be/api/shm/common/v1/">
   <message>cvc-complex-type.2.4.a: Invalid content was found starting with element
'tns:statusDate'. One of
```

```
'{&amp;quot;http://schema.post.be/api/galactic/order/v1/&amp;quot;:statusValue}' is
expected.</message>
   <timestamp>2011-04-27+02:00</timestamp>
</ns2:xsdValidationException>
```

6.1.6 Security

Authentication

Authentication is performed by the Server hosting the Web Services. We use pre-emptive Authentication over a secure channel: **HTTPS**.

This means the server will expect the **Authorization:** header to be sent along with the request. The value of this header is the authorization type (Basic) followed by the e-tailer's external-Id concatenated with a colon and an at least 128 bits pass-phrase (that will be communicated upon agreement with the external party). This concatenated value must be encoded in base64 before being actually set in the **Authorization:** header.

Authorization: Basic External-Id:pass-phrase
Where the underlined value is encoded in Base64. The External-Id is the e-tailer's Account ID.

For example, the Authorization Header for an e-tailer having the following attributes:
External-ID: **Etailer1**
Pass-Phrase: **QuiteLongPassPhrase**
Should generate an Authentication header and value string like:

Authorization: Basic RXRhaWxlcjE6UXVpdGVMb25nUGFzc1BocmFzZQ==

In case the call on the Web Service cannot be authenticated due to a missing or incorrect Authentication header, the caller of the Web Service will receive a response having an HTTP code 401 and an HTML body containing information similar to the example below:

```
HTTP/1.1 401 Unauthorized
Server: Apache-Coyote/1.1
Pragma: No-cache
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 01:00:00 CET
WWW-Authenticate: Basic realm="rest-realm"
Content-Type: text/html;charset=utf-8
Content-Length: 948
Date: Wed, 27 Apr 2011 14:33:36 GMT
Connection: close
<html>
   <head>
      <meta content="HTML Tidy for Java (vers. 26 sep 2004), see www.w3.org"
name="generator"/>
```

```
    <title>JBossWeb/2.0.1.GA - Error report</title>
    <style type="text/css">
    [...]
    </style>
  </head>
  <body>
    <h1>HTTP Status 401 -</h1>
    <hr noshade="noshade" size="1"/>
    <p>
      <b>type</b>
      Status report
    </p>
    <p>
      <b>message</b>
    </p>
    <p>
      <b>description</b>
      <u>This request requires HTTP authentication ().</u>
    </p>
    <hr noshade="noshade" size="1"/>
    <h3>JBossWeb/2.0.1.GA</h3>
  </body>
</html>
```

Authorization

Authorization is performed at the Web Service Level.

Each service end point contains the accountId to uniquely identify the resource. Prior to the accountId/passphrase validation, the web service will validate that the accountId found in the basic authentication and the accountId found in the resource are the same. This ensures a shop cannot access another shop's data.

If this is the case the Web Service will execute as described in the next sections of this document. If this is not the case the Web Service will not execute and the same response having HTTP code 401, as the example already provided above, will be sent back.

6.1.7 Status Codes

HTTP-based REST leverages the use of standard status codes:

- **4xx client error** status codes are used to map internal **Functional Exceptions**: The request can not be completed due to, for example, a conflict with the state of the resource: trying to confirm an order that has previously been cancelled, or trying to open an order referencing a product that doesn't exist in the catalog anymore. Another characteristic for these exceptions is that they can usually be solved by changing the content of the request the web service.

- **5xx server error** status codes are used to map internal **Technical Exceptions**: The request can not be completed due to an unexpected condition on the server side. For example, a failure connecting to the database or master data not being present in the database can be categorized as technical exceptions.

Functional Exceptions

If the Web Service call encounters a functional problem, a specific response will be sent back to the client. The response will be mapped to the most appropriate HTTP 1.1 status code in the 4xx range as defined here *http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html* and an XML content will be provided with a distinct error code and error message providing the client with a clear description of what went wrong as well as providing hints on what to change in order to solve the functional issue. A specific content-type is used to describe the version of the functional exception XML payload.

The following table summarizes the specific attributes of the functional exception response:

Attribute	Value/Description
HTTP Status	One of 400 Bad Request 403 Forbidden 404 Not Found 405 Method Not Allowed 406 Not Acceptable 409 Conflict 410 Gone 411 Length Required 415 Unsupported Media Type 416 Requested Range Not Satisfiable 417 Expectation Failed
HTTP Header	application/vnd.bpost.shmFunctionalException-v1+XML
HTTP Body	"businessException" as described in the Common-1.0.xsd

Below an example is provided of a response returned by any Web Service encountering a functional problem.

```
HTTP/1.1 409 Conflict
Server: Apache-Coyote/1.1
Content-Type: application/vnd.bpost.shmFunctionalException-v1+XML
Content-Length: 379
Date: Tue, 26 Apr 2011 07:30:20 GMT
Connection: close

<ns2:businessException xmlns="http://schema.post.be/common/exception/v1/"
xmlns:ns2="http://schema.post.be/api/shm/v1/">
   <code>409</code>
   <message>The order is in CANCELLED state and cannot be modified
anymore.</message>
</ns2:businessException>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

Technical Exceptions

If the Web Service call encounters a technical problem, a specific response will be sent back to the client. The response will be mapped to the most appropriate HTTP status code in the 5xx range. If the issue happened while the code of the web service is executed, the HTTP status code will always be 500 and XML code will be provided in the content of the response with a generic error message and a unique token used to uniquely identify the problem on our side. A specific content-type is used to describe the version of the functional exception XML payload.

The following table summarizes the specific attributes of the functional exception response:

Attribute	Value/Description
HTTP Status	One of: 500 Internal Server Error 501 Not Implemented 502 Bad Gateway 503 Service Unavailable 504 Gateway Timeout 505 HTTP Version Not Supported
HTTP Header	application/vnd.bpost.shmSystemException-v1+XML
HTTP Body	"systemException" as described in the Common-1.0.xsd

Below an example is provided of the response returned by any Web Service encountering a technical problem. The 500 Internal Server Error message contains a unique ID (UUID) that is important for technical support.

```
HTTP/1.1 500 Internal Server Error
Date: Fri, 29 Apr 2011 15:37:33 GMT
Server: Apache
Content-Length: 496
Connection: close
Content-Type: application/vnd.bpost.shmSystemException-v1+xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><systemException
xmlns="http://schema.post.be/api/shm/common/v2/"
xmlns:ns2="http://schema.post.be/common/exception/v1/"><ns2:message>An
unexpected error occurred while executing the request!
 Please try again in a few moments.
If the problem persist, please contact our support and provide the following token
information f35c0f13-538f-41ae-99aa-
932bc3141109</ns2:message><ns2:timestamp>2011-04-
29+02:00</ns2:timestamp></systemException>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

6.2 Create or Replace Order Web Service

The Create Order web service creates a new order. If an order with the same orderReference already exists, this service will act as a Replace Order web service, which cancels the existing order and creates a new one with the same orderReference.

6.2.1 Operation

To use the Create Order web service, you need to perform an HTTP operation on a URI that is constructed as follows:

URI: **serviceEndPoint/{accountId}/orders/**

Where
serviceEndPoint is **https://api.bpost.be/services/shm** and **{accountId}** is the same account number you use for authentication.

The only HTTP operation that is allowed on the Create Order URI is POST.

URI	POST	PUT	GET	DELETE
serviceEndPoint/{accountId}/orders/	☑	X	X	X

ATTENTION PUT, GET and DELETE operations on a Create Order URI are prohibited. Trying to perform these operations will always return a response with HTTP status code 405 Method Not Allowed.

6.2.2 Creating an order

When you want to create an order, you need to send the order information to the server using the HTTP POST operation on the URI. We will now show you how to send a valid request to create an order and what the response of the server will look like.

Client Request

Use the HTTP **POST** request method to send the order information to the server. The order information needs to be sent to the following **URL**:

Attribute	Value
HTTP Operation	POST
URL	https://api.bpost.be/services/shm/{accountId}/orders/

The HTTP POST request must contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-order-v2+XML

In the body of the HTTP POST request you need to put the XML code describing the order.

Attribute	Description
HTTP Body	XML <order/> element

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

<order> element tags

Name	Allowed Values	Description
accountId		Your unique customer ID of bpost P&E
orderReference		Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
status	value="OPEN" value="PENDING" value="CANCELLED" value="COMPLETED" value="ON-HOLD"	Order status
costCenter		This information is used on your invoice and allows you to attribute different cost centers
orderLine		The items that are included in the order. Order lines are shown in the back end of the shipping manager and facilitate the use of the tool. The subtags are explained in **<orderLine> element tags**.
customer		Customer tags. The subtags are explained in **<customer> element tags**.
deliveryMethod		Delivery method tags. The subtags are explained in **<deliveryMethod> element tags**.
totalPrice		Total price of the basket order in Euro cents (excluding shipping costs)

<orderLine> element tags

Name	Allowed Values	Description
text		Text describing the ordered item
nbOfItems		Number of items

<customer> element tags

Name	Allowed Values	Description
firstName		First name of the customer
lastName		Last name of the customer. **The first name and the last name**

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

		together should not be longer than 40 characters. Only 40 characters are shown on the label
deliveryAddress		Delivery address tags. The subtags are explained in **<customer> <deliveryAddress> element tags**.
email	Max length = 50	e-mail of the customer
phoneNumber	Max length = 20	Phone number of the customer

<customer> <deliveryAddress> element tags

Name	Allowed Values	Description
streetName	Max length = 40	Street name of the customer
number	Max length = 8	Street number of the customer
box	Max length = 8	Box number of the customer
postalCode	Max length = 40	Postal code of the customer
locality	Integer Max length = 40	City name of the customer
countryCode	Uppercase 2 character country code	Country of the customer

<deliveryMethod> element tags

Name	Allowed Values	Description
atHome		@home tags. The subtags are explained in **<deliveryMethod> <atHome> element tags**.
atShop		@shop tags. The subtags are explained in **<deliveryMethod> <atShop> element tags**.
at24-7		Bpack24/7 tags. The subtags are explained in **<deliveryMethod> <at24-7> element tags**.
intExpress		International Express tags.
intBusiness		International Business tags.

<deliveryMethod> <atHome> element tags

Name	Allowed Values	Description
normal		Normal @home. The subtags are explained in **<deliveryMethod> <atHome> <normal> element tags**.

signed		@home + signature The subtags are explained in **<deliveryMethod> <atHome> <signed> element tags**.
insured		@home + insurance. The subtags are explained in **<deliveryMethod> <atHome> <insured> element tags**.
dropAtTheDoor		Drop parcel at the door of customer

<deliveryMethod> <atHome> <normal> element tags

Name	Allowed Values	Description
options		For the options subtags see **Additional services: <options> element tags**.

<deliveryMethod> <atHome> <signed> element tags

Name	Allowed Values	Description
"Signature tag"		One of the subtags that are used under the <signature> tag. See **<signature> element tags**.
options		For the options subtags see **Additional services: <options> element tags**.

<deliveryMethod> <atHome> <insured> element tags

Name	Allowed Values	Description
"Insurance tag"		One of the subtags that are used under the <insurance> tag. See **<insurance> element tags**.
options		For the options subtags see **Additional services: <options> element tags**.

<deliveryMethod> <atShop> element tags

Name	Allowed Values	Description
infoPugo		@shop Pick Up & Go tags. The subtags are explained in **<infoPugo> element tags**.
insurance		@shop + insurance. The subtags are explained in **<insurance> element tags**.
infoDistributed		Info distributed. The subtags are

		explained in **Notification tags**.

<deliveryMethod> <atShop> <infoPugo> element tags

Name	Allowed Values	Description
pugoId		Pick Up & Go identifier
pugoName		Pick Up & Go name
"Notification tag"		One of the notification tags. See **Notification tags**.

<deliveryMethod> <at24-7> element tags

Name	Allowed Values	Description
infoParcelsDepot		Bpack24/7 Parcels depot tags. The subtags are explained in **<deliveryMethod> <at24-7> <infoParcelsDepot> element tags**.
signature		Bpack24/7 + signature. The subtags are explained in **<signature> element tags**.
insurance		Bpack24/7 + insurance. The subtags are explained in **<insurance> element tags**.
memberId		Bpack24/7 member identifier

<deliveryMethod> <at24-7> <infoParcelsDepot> element tags

Name	Allowed Values	Description
parcelsDepotId		Parcels depot identifier

<deliveryMethod> <intExpress> element tags

Name	Allowed Values	Description
insured		International Express + insurance. The subtags are explained in **<deliveryMethod> <intExpress> <insured> element tags**.

<deliveryMethod> <intExpress> <insured> element tags

Name	Allowed Values	Description
"Insurance tag"		One of the subtags that are used under the <insurance> tag. See **<insurance> element tags**.
options		For the options subtags see **Additional services: <options> element tags**.

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

<deliveryMethod> <intBusiness> element tags

Name	Allowed Values	Description
insured		International Business + insurance. The subtags are explained in **<deliveryMethod> <intBusiness> <insured> element tags**.

<deliveryMethod> <intBusiness> <insured> element tags

Name	Allowed Values	Description
"Insurance tag"		One of the subtags that are used under the <insurance> tag. See **<insurance> element tags**.
options		For the options subtags see **Additional services: <options> element tags**.

Additional services: <options> element tags

Name	Allowed Values	Description
infoDistributed		Info distributed. The subtags are explained in **Notification tags**.
infoNextDay		Info next day. The subtags are explained in **Notification tags**.
infoReminder		Info reminder. The subtags are explained in **Notification tags**.
automaticSecondPresentation		Automatic second presentation

Notification tags

Name	Attributes	Description
emailAddress	language	e-mail address if you want to notify by e-mail. The language attribute is required. See **Notification language attribute**.
mobilePhone	language	Mobile phone number if you want to notify by mobile phone. The language attribute is required. See **Notification language attribute**.
fixedPhone	language	Fixed phone number if you want to notify by fixed phone. The language attribute is

		required. See **Notification language attribute**.

Notification language attribute

Name	Allowed Values	Description
language	EN NL FR DE	The language of the notification. EN = English NL = Dutch FR= French DE = German

<signature> element tags

Name	Allowed Values	Description
signature		Delivery method with signature
signaturePlus		Delivery method with signature, name and date

<insurance> element tags

Name	Attributes	Description
basicInsurance		Basic insurance
additionalInsurance	value="1" value="2" value="3" value="4" value="5" value="6" value="7" value="8" value="9" value="10" value="11"	The range in which the insurance amount is situated: 1 = basic insurance up to 500 EUR 2 = additional up to 2.500EUR 3 = additional up to 5.000 EUR 4 = additional up to 7.500 EUR 5 = additional up to 10.000 EUR 6 = additional up to 12.500 EUR 7 = additional up to 15.000 EUR 8 = additional up to 17.500 EUR 9 = additional up to 20.000 EUR 10 = additional up to 22.500 EUR 11 = additional up to 25.000 EUR

Example 1: @home

The following example shows a valid request to create an order with delivery method @home:

```
POST /shm/123456/orders
Content-type: application/vnd.bpost.shm-order-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<order xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <accountId>123456</accountId>
   <orderReference>201106241506</orderReference>
   <status>OPEN</status>
   <costCenter>costCenter0</costCenter>
   <orderLine>
      <text>Iphone 4 16GB Black</text>
      <nbOfItems>1</nbOfItems>
   </orderLine>
   <orderLine>
      <text>Earphones</text>
      <nbOfItems>1</nbOfItems>
   </orderLine>
   <customer>
      <firstName>Jan</firstName>
      <lastName>Peeters</lastName>
      <deliveryAddress>
         <streetName>Dorpsstraat</streetName>
         <number>132</number>
         <box></box>
         <postalCode>1800</postalCode>
         <locality>Vilvoorde</locality>
         <countryCode>BE</countryCode>
      </deliveryAddress>
      <email>jan.peeters@provider.be</email>
      <phoneNumber>0032499123456</phoneNumber>
   </customer>
   <deliveryMethod>
      <atHome>
         <normal/>
      </atHome>
   </deliveryMethod>
   <totalPrice>66490</totalPrice>
</order>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

Example 2: @shop

The following example shows a valid request to create an order with delivery method @shop:

```
POST /shm/123456/orders
Content-type: application/vnd.bpost.shm-order-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<order xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <accountId>123456</accountId>
   <orderReference>201106241506</orderReference>
   <status>OPEN</status>
   <costCenter>costCenter0</costCenter>
   <orderLine>
      <text>Iphone 4 16GB Black</text>
      <nbOfItems>1</nbOfItems>
   </orderLine>
   <orderLine>
      <text>Earphones</text>
      <nbOfItems>1</nbOfItems>
   </orderLine>
   <customer>
      <firstName>Jan</firstName>
      <lastName>Peeters</lastName>
      <deliveryAddress>
         <streetName>Dorpsstraat</streetName>
         <number>132</number>
         <box></box>
         <postalCode>1800</postalCode>
         <locality>Vilvoorde</locality>
         <countryCode>BE</countryCode>
      </deliveryAddress>
      <email>jan.peeters@provider.be</email>
      <phoneNumber>0032499123456</phoneNumber>
   </customer>
   <deliveryMethod>
      <atShop>
         <infoPugo>
            <pugoId>pugoId0</pugoId>
            <pugoName>pugoName0</pugoName>
            <emailAddress language=NL>jan.peeters@provider.be</emailAddress>
         </infoPugo>
         <insurance>
            <additionalInsurance value="2"/>
         </insurance>
         <infoDistributed>
            <emailAddress language=NL>jan.peeters@provider.be</emailAddress>
         </infoDistributed>
      </atShop>
   </deliveryMethod>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
   <totalPrice>66490</totalPrice>
</order>
```

Server Response

If your request to create an order is successful, the server will respond with an HTTP **200 OK** status code.

Attribute	Value
HTTP Status	200 OK

The server will give you the location of the order you created as one of the header fields:

Attribute	Value
HTTP Header	Location: https://api.bpost.be/services/shm/{accountId}/orders/{orderReference}

The body of the response message will be empty, because there is no XML code that needs to be sent back.

Attribute	Description
HTTP Body	empty

Example

The following example shows a response confirming that the order was created successfully:

```
HTTP/1.1 200 OK
Location: https://api.bpost.be/services/shm/123456/orders/201106241506
```

6.3 Fetch Order Web Service

The Fetch Order web service retrieves an order by its orderReference. If there are multiple orders that match the provided orderReference, only the most recent order that was not cancelled is returned. Barcodes that are attached to the requested orderReference will be returned.

6.3.1 Operation

To use the Fetch Order web service, you need to perform an HTTP operation on a URI that is constructed as follows:

URI: **serviceEndPoint/{accountId}/orders/{orderReference}**

Where
serviceEndPoint is **https://api.bpost.be/services/shm** and **{accountId}** is the same account number you use for authentication.

The only HTTP operation that is allowed on the Fetch Order URI is GET.

URI	POST	PUT	GET	DELETE
serviceEndPoint/{accountId}/orders/{orderReference}	X	X	☑	X

ATTENTION POST, PUT and DELETE operations on a Fetch Order URI are prohibited. Trying to perform these operations will always return a response with HTTP status code 405 Method Not Allowed.

6.3.2 Fetching an order

When you want to fetch an order, you need to send a request to the server to receive order information using the HTTP GET operation on the URI. We will now show you how to send a valid request to fetch an order and what the response of the server will look like.

Client Request

Use the HTTP **GET** request method to fetch order information. To retrieve Order information you need to request it from the following **URL**:

Attribute	Value
HTTP Operation	GET
URL	https://api.bpost.be/services/shm/{accountId}/orders/{orderReference}

The HTTP GET request must contain an **Accept** header field:

Attribute	Value
HTTP Header	Accept: application/vnd.bpost.shm-order-v2+XML

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

The body of the GET request will be empty, because there is no XML code that needs to be sent to the server.

Attribute	Description
HTTP Body	empty

Example
The following example shows a valid request to fetch an order:

```
GET /shm/123456/orders/201106241506
Accept: application/vnd.bpost.shm-order-v2+XML
```

Server Response

If your request to fetch an order is successful, the server will respond with an HTTP **200 OK** status code.

Attribute	Value
HTTP Status	200 OK

The server response will contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-order-v2+XML

In the body of the server response you will receive XML code containing the orderWithBarcodes element.

Attribute	Description
HTTP Body	XML <orderWithBarcodes/> element

The <orderWithBarcodes/> element will contain the order and if they have been created the barcodes attached to the order.

<orderWithBarcodes> element tags

Name	Allowed Values	Description
order		<order> element tags are explained in **6.2 Create or Replace Order Web Service**
barcode		Barcode.

If barcodes are not (yet) attached, the element will not be included.

Example
The following example shows a response by the server for a Bpack@bpost (PUGO) order.

```
HTTP/1.1 200 OK
Content-type: application/vnd.bpost.shm-order-v2+XML
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
<?xml version="1.0" encoding="UTF-8"?>
<orderWithBarcodes xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <order>
      <accountId>123456</accountId>
      <orderReference>201106241506</orderReference>
      <status>OPEN</status>
      <costCenter>costCenter0</costCenter>
      <orderLine>
         <text>Iphone 4 16GB Black</text>
         <nbOfItems>1</nbOfItems>
      </orderLine>
      <orderLine>
         <text>Earphones</text>
         <nbOfItems>1</nbOfItems>
      </orderLine>
      <customer>
         <firstName>Jan</firstName>
         <lastName>Peeters</lastName>
         <deliveryAddress>
            <streetName>Dorpsstraat</streetName>
            <number>132</number>
            <box></box>
            <postalCode>1800</postalCode>
            <locality>Vilvoorde</locality>
            <countryCode>BE</countryCode>
         </deliveryAddress>
         <email>jan.peeters@provider.be</email>
         <phoneNumber>0032499123456</phoneNumber>
      </customer>
      <deliveryMethod>
         <atShop>
            <infoPugo>
               <pugoId>pugoId0</pugoId>
               <pugoName>pugoName0</pugoName>
               <emailAddress language=NL>jan.peeters@provider.be</emailAddress>
            </infoPugo>
            <insurance>
               <additionalInsurance value="1"/>
            </insurance>
            <infoDistributed>
               <emailAddress language=NL>jan.peeters@provider.be</emailAddress>
            </infoDistributed>
         </atShop>
      </deliveryMethod>
      <totalPrice>66490</totalPrice>
   </order>
   <barcode>32321234565990000000001030</barcode>
</orderWithBarcodes>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

Example
The following example shows a response by the server for a Bpack 24/7 order.

```
<?xml version="1.0" encoding="UTF-8"?>
<orderWithBarcodes xmlns="http://schema.post.be/shm/deepintegration/v2/"
xmlns:ns2="http://schema.post.be/common/exception/v1/">
   <order>
      <accountId>123456</accountId>
      <orderReference>be</orderReference>
      <status>PENDING</status>
      <customer>
         <firstName>Ts</firstName>
         <lastName>Gs</lastName>
         <deliveryAddress>
            <streetName>PB20002 AÃ©roport Zaventem</streetName>
            <number>17</number>
            <postalCode>1930</postalCode>
            <locality>Zaventem</locality>
            <countryCode>BE</countryCode>
         </deliveryAddress>
      </customer>
      <deliveryMethod>
         <at24-7>
            <infoParcelsDepot>
               <parcelsDepotId>014473</parcelsDepotId>
            </infoParcelsDepot>
            <memberId>123456789</memberId>
         </at24-7>
      </deliveryMethod>
      <totalPrice>500</totalPrice>
   </order>
   <barcode>32329990105990000037903O</barcode>
</orderWithBarcodes>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

6.4 Modify Order Status Web Service

The Modify Order Status web service modifies the status of an order.

6.4.1 Operation

To use the Modify Order Status web service, you need to perform an HTTP operation on a URI that is constructed as follows:

URI: **serviceEndPoint/{accountId}/orders/status**

Where
serviceEndPoint is **https://api.bpost.be/services/shm** and **{accountId}** is the same account number you use for authentication.

The only HTTP operation that is allowed on the Modify Order Status URI is PUT.

URI	POST	PUT	GET	DELETE
serviceEndPoint/{accountId}/orders/status	X	☑	X	X

ATTENTION POST, GET and DELETE operations on a Modify Order Status URI are prohibited. Trying to perform these operations will always return a response with HTTP status code 405 Method Not Allowed.

6.4.2 Modifying an order status

When you want to modify an order status, you need to send the updated order status to the server using the HTTP PUT operation on the URI. We will now show you how to send a valid request to modify an order status and what the response of the server will look like.

ATTENTION Modifying the status of a cancelled order is not allowed.

Client Request

Use the HTTP **PUT** request method to modify the order status. The updated status needs to be sent to the following **URL**:

Attribute	Value
HTTP Operation	PUT
URL	https://api.bpost.be/services/shm/{accountId}/orders/status

The HTTP PUT request must contain the **Content-type** and **X-HTTP-Method-Override** header fields:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-order-status-v2+XML
HTTP Header	X-HTTP-Method-Override: PATCH

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

In the body of the HTTP PUT request you need to put the XML code to update the order status.

Attribute	Description
HTTP Body	XML <orderStatusMap/> element

<orderStatusMap> element tags

Name	Allowed Values	Description
orderReference		Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
status	value="OPEN" value="PENDING" value="CANCELLED" value="COMPLETED" value="ON-HOLD	

Example
The following example shows a valid request to modify an order:

```
PUT /shm/123456/orders/status
Content-type: application/vnd.bpost.shm-order-status-v2+XML
X-HTTP-Method-Override: PATCH

<?xml version="1.0" encoding="UTF-8"?>
<orderStatusMap xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <entry>
      <orderReference>201106241506</orderReference>
      <status>PENDING</status>
   </entry>
</orderStatusMap>
```

Server Response

If your request to modify an order status is successful, the server will respond with an HTTP **200 OK** status code.

Attribute	Value
HTTP Status	200 OK

The server will give you the location of the updated order as one of the header fields:

Attribute	Value
HTTP Header	Location: https://api.bpost.be/services/shm/{accountId}/orders/{orderReference}

The body of the response message will be empty, because there is no XML code that needs to be sent back.

Attribute	Description
HTTP Body	empty

Example

The following example shows a response confirming that the order status was modified successfully:

```
HTTP/1.1 200 OK
Location: https://api.bpost.be/services/shm/123456/orders/201106241506
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

6.5 Create National Labels Web Service

The Create National Labels web service creates new national labels.

6.5.1 Operation

To use the Create National Labels web service, you need to perform an HTTP operation on a URI that is constructed as follows:

URI: **serviceEndPoint/{accountId}/labels/**

Where
serviceEndPoint is **https://api.bpost.be/services/shm** and **{accountId}** is the same account number you use for authentication.

The only HTTP operation that is allowed on the Create National Labels URI is POST.

URI	POST	PUT	GET	DELETE
serviceEndPoint/{accountId}/labels/	☑	X	X	X

ATTENTION PUT, GET and DELETE operations on a Create National Labels URI are prohibited. Trying to perform these operations will always return a response with HTTP status code 405 Method Not Allowed.

6.5.2 Creating a national label

When you want to create a national label, you need to send the national label information to the server using the HTTP POST operation on the URI. We will now show you how to send a valid request to create a national label and what the response of the server will look like.

Client Request

Use the HTTP **POST** request method to send the national label information to the server. The label information needs to be sent to one of the following **URLs**:

Attribute	Value
HTTP Operation	POST
URL	https://api.bpost.be/services/shm/{accountId}/labels/
URL	https://api.bpost.be/services/shm/{accountId}/labels?labelFormat=A_4
URL	https://api.bpost.be/services/shm/{accountId}/labels?labelFormat=A_5

By default, labels requested using the first URL will be printed on an A4 page. Labels requested using the second URL with the labelFormat=A_4 specification will also be printed on an A4 page. One A4 page can fit 4 labels. If you want to print each label separately on its own A5 size page, you need to use the labelFormat=A_5 specification in the URL.

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

The HTTP POST request must contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-nat-label-v2+XML

In the body of the HTTP POST request you need to put the XML code containing the label information.

Attribute	Description
HTTP Body	XML <orderRefLabelAmountMap/> element

<orderRefLabelAmountMap> element tags

Name	Allowed Values	Description
orderReference		Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
labelAmount		Amount of labels
withRetour	true false	Flag indicating whether return labels are included. When using the withRetour element set to true, the sender address and receiver address are switched. **This is only possible for @home delivery.** In case the person using the retour label has chosen a delivery method with an address not set as his home address (e.g. @shop or Bpack 24/7), the address cannot be used for the retour label. Therefore it is (currently) not possible to generate a retour label for those shipments.
returnLabels	0 1	Flag indicating whether labels are returned. 0=no 1=yes

Example

The following example shows a valid request to create a national label:

```
POST /shm/123456/labels/
Content-type: application/vnd.bpost.shm-nat-label-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<orderRefLabelAmountMap xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <entry>
      <orderReference>201106241506</orderReference>
      <labelAmount>4</labelAmount>
      <withRetour>false</withRetour>
      <returnLabels>0</returnLabels>
   </entry>
```

```
</orderRefLabelAmountMap>
```

Server Response

If your request to create a label is successful, the server will respond with an HTTP **200 OK** status code.

Attribute	Value
HTTP Status	200 OK

The server response will contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-nat-label-v2+XML

In the body of the server response you will receive XML code containing the barcode element.

Attribute	Description
HTTP Body	XML <orderRefBarcodeMap/> element

<orderRefBarcodeMap> element tags

Name	Allowed Values	Description
orderReference		Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
barcode		Bar code

Example

The following example shows a response confirming that the label was created successfully:

```
HTTP/1.1 200 OK
Content-type: application/vnd.bpost.shm-nat-label-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<orderRefBarcodeMap xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <entry>
      <orderReference>201106241506</orderReference>
      <barcode>3232123456599000000001030</barcode>
   </entry>
</orderRefBarcodeMap>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

6.6 Create International Labels Web Service

The Create International Labels web service creates new international labels.

6.6.1 Operation

To use the Create International Labels web service, you need to perform an HTTP operation on a URI that is constructed as follows:

URI: **serviceEndPoint/{accountId}/labels/**

Where
serviceEndPoint is **https://api.bpost.be/services/shm** and **{accountId}** is the same account number you use for authentication.

The only HTTP operation that is allowed on the Create International Labels URI is POST.

URI	POST	PUT	GET	DELETE
serviceEndPoint/{accountId}/labels/	☑	X	X	X

ATTENTION PUT, GET and DELETE operations on a Create International Labels URI are prohibited. Trying to perform these operations will always return a response with HTTP status code 405 Method Not Allowed.

6.6.2 Creating an international label

When you want to create an international label, you need to send the international label information to the server using the HTTP POST operation on the URI. We will now show you how to send a valid request to create an international label and what the response of the server will look like.

Client Request

Use the HTTP **POST** request method to send the international label information to the server. The label information needs to be sent to one of the following **URL**:

Attribute	Value
HTTP Operation	POST
URL	https://api.bpost.be/services/shm/{accountId}/labels/

International labels will be printed on an A4 page. One A4 page can fit 4 labels.

The HTTP POST request must contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-int-label-v2+XML

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

In the body of the HTTP POST request you need to put the XML code containing the international label information.

Attribute	Description
HTTP Body	XML <internationalLabelInfos/> element

<internationalLabelInfos> element tags

Name	Allowed Values	Description
internationalLabelInfo		For each label the <internationalLabelInfo> tag needs to be present. The subtags are explained in **<internationalLabelInfo> element tags**.
orderReference		Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
returnLabels	true false	Flag indicating whether labels are returned.

<internationalLabelInfo> element tags

Name	Allowed Values	Description
parcelValue		Value of the parcel in Euro cent.
parcelWeight		Weight of the parcel in grams.
contentDescription		Content description
shipmentType	SAMPLE GIFT OTHER DOCUMENTS	Shipment type
parcelReturnInstructions	RTA ABANDONED RTS	Return instructions
privateAddress	true false	Flag indicating whether the address is a private address (true) or a business address (false).

Example

The following example shows a valid request to create international labels. In the example there are two <internationalLabelInfo> blocks, which means this code will create two international labels for one order:

```
POST /shm/123456/labels/
Content-type: application/vnd.bpost.shm-int-label-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<internationalLabelInfos xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <internationalLabelInfo>
      <parcelValue>990</parcelValue>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
      <parcelWeight>250</parcelWeight>
      <contentDescription>My description</contentDescription>
      <shipmentType>OTHER</shipmentType>
      <parcelReturnInstructions>RTA</parcelReturnInstructions>
      <privateAddress>true</privateAddress>
   </internationalLabelInfo>
   <internationalLabelInfo>
      <parcelValue>300</parcelValue>
      <parcelWeight>50</parcelWeight>
      <contentDescription>This is a gift</contentDescription>
      <shipmentType>GIFT</shipmentType>
      <parcelReturnInstructions>RTA</parcelReturnInstructions>
      <privateAddress>false</privateAddress>
   </internationalLabelInfo>
   <orderReference>201108191553</orderReference>
   <returnLabels>true</returnLabels>
</internationalLabelInfos>
```

Server Response

If your request to create an international label is successful, the server will respond with an HTTP **200 OK** status code.

Attribute	Value
HTTP Status	200 OK

The server response will contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-int-label-v2+XML

In the body of the server response you will receive XML code containing the barcode element.

Attribute	Description
HTTP Body	XML <orderRefBarcodeMap/> element

<orderRefBarcodeMap> element tags

Name	Allowed Values	Description
orderReference		Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
barcode		Bar code

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

Example

The following example shows a response confirming that the label was created successfully:

```
HTTP/1.1 200 OK
Content-type: application/vnd.bpost.shm-int-label-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<orderRefBarcodeMap xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <entry>
      <orderReference>201108191553</orderReference>
      <barcode>CD100000016BE</barcode>
   </entry>
</orderRefBarcodeMap>
```

6.7 Create Order And National Labels Web Service

The Create Order And National Labels web service creates an order and a certain amount of national labels in one call.

6.7.1 Operation

To use the Create Order And National Labels web service, you need to perform an HTTP operation on a URI that is constructed as follows:

URI: **serviceEndPoint/{accountId}/orderAndLabels/**

Where
serviceEndPoint is **https://api.bpost.be/services/shm** and **{accountId}** is the same account number you use for authentication.

The only HTTP operation that is allowed on the Create Order And National Labels URI is POST.

URI	POST	PUT	GET	DELETE
serviceEndPoint/{accountId}/orderAndLabels/	☑	X	X	X

ATTENTION PUT, GET and DELETE operations on a Create Order And National Labels URI are prohibited. Trying to perform these operations will always return a response with HTTP status code 405 Method Not Allowed.

6.7.2 Creating an order and a national label

When you want to create an order and a national label, you need to send the order and label information to the server using the HTTP POST operation on the URI. We will now show you how to send a valid request to create an order and a national label and what the response of the server will look like.

Client Request

Use the HTTP **POST** request method to send the order and label information to the server. The order and label information needs to be sent to the following **URL**:

Attribute	Value
HTTP Operation	POST
URL	https://api.bpost.be/services/shm/{accountId}/orderAndLabels/

The HTTP POST request must contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-orderAndNatLabels-v2+XML

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

In the body of the HTTP POST request you need to put the XML code containing the order and label information.

Attribute	Description
HTTP Body	XML <orderWithLabelAmount/> element

<orderWithLabelAmount> element tags

Name	Allowed Values	Description
order		<order> element tags are explained in **6.2 Create or Replace Order Web Service**
labelAmount		Amount of labels

Example

The following example shows a valid request to create an order and a national label:

```
POST shm/123456/orderAndLabels/
Content-type: application/vnd.bpost.shm-orderAndNatLabels-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<orderWithLabelAmount xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <order>
      <accountId>123456</accountId>
      <orderReference>201106241506</orderReference>
      <status>OPEN</status>
      <costCenter>costCenter0</costCenter>
      <orderLine>
         <text>Iphone 4 16GB Black</text>
         <nbOfItems>1</nbOfItems>
      </orderLine>
      <orderLine>
         <text>Earphones</text>
         <nbOfItems>1</nbOfItems>
      </orderLine>
      <customer>
         <firstName>Jan</firstName>
         <lastName>Peeters</lastName>
         <deliveryAddress>
            <streetName>Dorpsstraat</streetName>
            <number>132</number>
            <box></box>
            <postalCode>1800</postalCode>
            <locality>Vilvoorde</locality>
            <countryCode>BE</countryCode>
         </deliveryAddress>
         <email>jan.peeters@provider.be</email>
         <phoneNumber>0032499123456</phoneNumber>
      </customer>
      <deliveryMethod>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
        <atHome>
        </atHome>
      </deliveryMethod>
   </order>
   <labelAmount>4</labelAmount>
</orderWithLabelAmount>
```

Server Response

If your request to create an order and a national label is successful, the server will respond with an HTTP **200 OK** status code.

Attribute	Value
HTTP Status	200 OK

The server response will contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-label-v2+XML

In the body of the server response you will receive XML code containing the order and label element.

Attribute	Description
HTTP Body	XML <orderRefBarcodeMap/> element

<orderRefBarcodeMap> element tags

Name	Allowed Values	Description
orderReference		Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
barcode		Bar code

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

Example

The following example shows a response confirming that the label was created successfully:

```
HTTP/1.1 200 OK
Content-type: application/vnd.bpost.shm-label-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<orderRefBarcodeMap xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <entry>
      <orderReference>201106241506</orderReference>
      <barcode>32321234565990000000001030</barcode>
   </entry>
</orderRefBarcodeMap>
```

6.8 Create Order And International Labels Web Service

The Create Order And International Labels web service creates an order and a certain amount of international labels in one call.

6.8.1 Operation

To use the Create Order And International Labels web service, you need to perform an HTTP operation on a URI that is constructed as follows:

URI: **serviceEndPoint/{accountId}/orderAndLabels/**

Where
serviceEndPoint is **https://api.bpost.be/services/shm** and **{accountId}** is the same account number you use for authentication.

The only HTTP operation that is allowed on the Create Order And International Labels URI is POST.

URI	POST	PUT	GET	DELETE
serviceEndPoint/{accountId}/orderAndLabels/	☑	X	X	X

ATTENTION PUT, GET and DELETE operations on a Create Order And International Labels URI are prohibited. Trying to perform these operations will always return a response with HTTP status code 405 Method Not Allowed.

6.8.2 Creating an order and an international label

When you want to create an order and a international label, you need to send the order and label information to the server using the HTTP POST operation on the URI. We will now show you how to send a valid request to create an order and a international label and what the response of the server will look like.

Client Request

Use the HTTP **POST** request method to send the order and label information to the server. The order and label information needs to be sent to the following **URL**:

Attribute	Value
HTTP Operation	POST
URL	https://api.bpost.be/services/shm/{accountId}/orderAndLabels/

The HTTP POST request must contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-orderAndIntLabels-v2+XML

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

In the body of the HTTP POST request you need to put the XML code containing the order and label information.

Attribute	Description
HTTP Body	XML <orderInternationalLabelInfos/> element

<orderInternationalLabelInfos> element tags

Name	Allowed Values	Description
internationalLabelInfo		For each label the <internationalLabelInfo> tag needs to be present. The subtags are explained in **<internationalLabelInfo> element tags**.
order		<order> element tags are explained in **6.2 Create or Replace Order Web Service**.

<internationalLabelInfo> element tags

Name	Allowed Values	Description
parcelValue		Value of the parcel in Euro cent.
parcelWeight		Weight of the parcel in grams.
contentDescription		Content description
shipmentType	SAMPLE GIFT OTHER DOCUMENTS	Shipment type
parcelReturnInstructions	RTA ABANDONED RTS	Return instructions
privateAddress	true false	Flag indicating whether the address is a private address (true) or a business address (false).

Example

The following example shows a valid request to create an order and a national label:

```
POST shm/123456/orderAndLabels/
Content-type: application/vnd.bpost.shm-orderAndIntLabels-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<orderInternationalLabelInfos
xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <internationalLabelInfo>
      <parcelValue>66490</parcelValue>
      <parcelWeight>500</parcelWeight>
      <contentDescription>Iphone 4 16GB Black + Earphones</contentDescription>
      <shipmentType>OTHER</shipmentType>
      <parcelReturnInstructions>RTA</parcelReturnInstructions>
      <privateAddress>true</privateAddress>
   </internationalLabelInfo>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
  <order>
    <accountId>123456</accountId>
    <orderReference>201106241506</orderReference>
    <status>OPEN</status>
    <costCenter>costCenter0</costCenter>
    <orderLine>
      <text>Iphone 4 16GB Black</text>
      <nbOfItems>1</nbOfItems>
    </orderLine>
    <orderLine>
      <text>Earphones</text>
      <nbOfItems>1</nbOfItems>
    </orderLine>
    <customer>
      <firstName>Jan</firstName>
      <lastName>Peeters</lastName>
      <deliveryAddress>
        <streetName>Dorpsstraat</streetName>
        <number>132</number>
        <box></box>
        <postalCode>1800</postalCode>
        <locality>Vilvoorde</locality>
        <countryCode>BE</countryCode>
      </deliveryAddress>
      <email>jan.peeters@provider.be</email>
      <phoneNumber>0032499123456</phoneNumber>
    </customer>
    <deliveryMethod>
      <intExpress>
      </intExpress>
    </deliveryMethod>
  </order>
</orderInternationalLabelInfos>
```

Server Response

If your request to create an order and a national label is successful, the server will respond with an HTTP **200 OK** status code.

Attribute	Value
HTTP Status	200 OK

The server response will contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-orderAndIntLabels-v2+XML

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

In the body of the server response you will receive XML code containing the order and label element.

Attribute	Description
HTTP Body	XML <orderRefBarcodeMap/> element

<orderRefBarcodeMap> element tags

Name	Allowed Values	Description
orderReference		Order reference: unique ID used in your web shop to assign to an order. The value of this parameter is not managed by bpost. **If the value already exists, it will overwrite current order info.**
barcode		Bar code

Example

The following example shows a response confirming that the label was created successfully:

```
HTTP/1.1 200 OK
Content-type: application/vnd.bpost.shm-orderAndIntLabels-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<orderRefBarcodeMap xmlns="http://schema.post.be/shm/deepintegration/v2/">
   <entry>
      <orderReference>201106241506</orderReference>
      <barcode>CD100000016BE</barcode>
   </entry>
</orderRefBarcodeMap>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

6.9 Retrieve PDF Labels For Box Web Service

The Retrieve PDF Labels For Box web service retrieves printable barcodes in a PDF file. This web service will retrieve both national and international labels.

6.9.1 Operation

To use the Retrieve PDF Labels For Box web service, you need to perform an HTTP operation on a URI that is constructed as follows:

URI: **serviceEndPoint/{accountId}/orders/{orderReference}**

Where
serviceEndPoint is **https://api.bpost.be/services/shm** and **{accountId}** is the same account number you use for authentication.

The only HTTP operation that is allowed on the Retrieve Barcode as PDF URI is GET.

URI	POST	PUT	GET	DELETE
serviceEndPoint/{accountId}/labels/{barcode}/pdf	X	X	☑	X

ATTENTION POST, PUT and DELETE operations on a Retrieve PDF Labels For Box URI are prohibited. Trying to perform these operations will always return a response with HTTP status code 405 Method Not Allowed.

6.9.2 Retrieving a PDF label for a box

When you want to retrieve a PDF label for a box, you need to send a request to the server to receive a PDF file containing the barcode using the HTTP GET operation on the URI. We will now show you how to send a valid request to retrieve a barcode as PDF (base64 format) and what the response of the server will look like.

Client Request

Use the HTTP **GET** request method to retrieve a barcode as PDF from one of the following **URLs**:

Attribute	Value
HTTP Operation	GET
URL	https://api.bpost.be/services/shm/{accountId}/labels/{barcode}/pdf
URL	https://api.bpost.be/services/shm/{accountId}/labels/{barcode}/pdf?labelFormat=A_4
URL	https://api.bpost.be/services/shm/{accountId}/labels/{barcode}/pdf?labelFormat=A_5

By default, labels requested using the first URL will be printed on an A4 page. Labels requested using the second URL with the labelFormat=A_4 specification will also be printed on an A4 page. One A4 page can fit 4 labels. If you want to print each label separately on its own A5 size page, you need to use the labelFormat=A_5 specification in the URL. The labelFormat parameter doesn't change the format of international labels, because only the A4 format is supported for international labels.

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

The HTTP GET request must contain an **Accept** header field:

Attribute	Value
HTTP Header	Accept: application/vnd.bpost.shm-pdf-v2+XML

The body of the GET request will be empty, because there is no XML code that needs to be sent to the server.

Attribute	Description
HTTP Body	empty

Example
The following example shows a valid request to retrieve a barcode as PDF:

```
GET shm/123456/labels/323212345659900000001030/pdf
Accept: application/vnd.bpost.shm-pdf-v2+XML
```

Server Response

If your request to retrieve a barcode as PDF is successful, the server will respond with an HTTP **200 OK** status code.

Attribute	Value
HTTP Status	200 OK

The server response will contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-pdf-v2+XML

In the body of the server response you will receive XML code containing the PDF element.

Attribute	Description
HTTP Body	XML <pdf/> element

Example
The following example shows a response by the server giving you the barcode as PDF. The PDF file will be encoded in base64. Only a part of the code is shown here as an example:

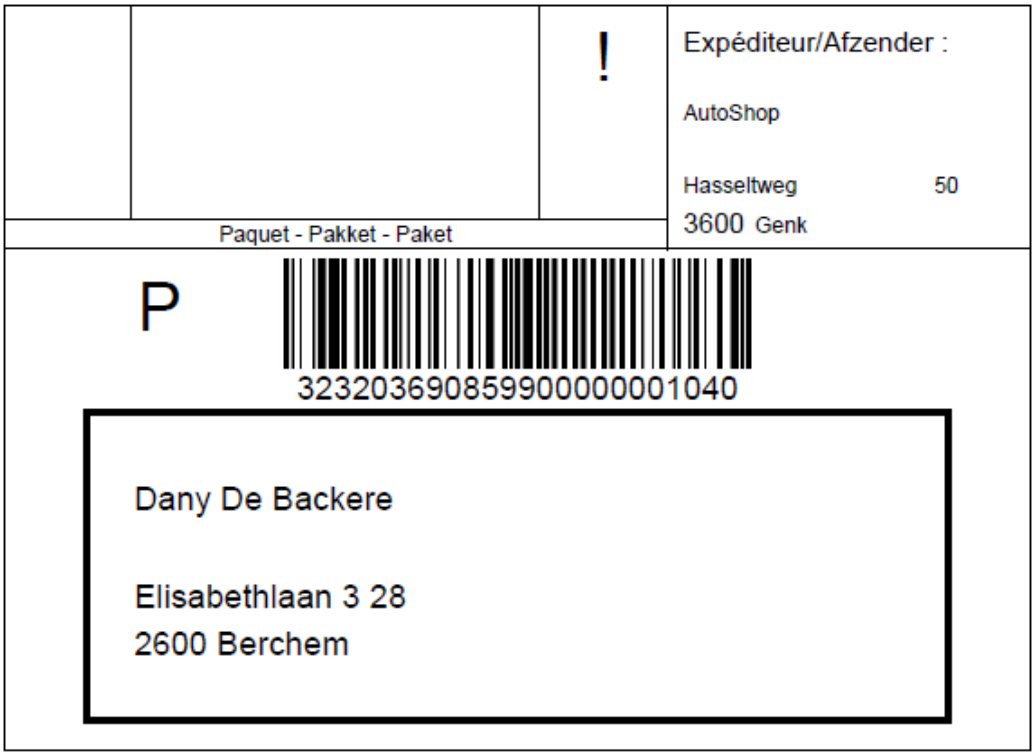
```
HTTP/1.1 200 OK
Content-type: application/vnd.bpost.shm-pdf-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<pdf xmlns="http://schema.post.be/shm/deepintegration/v2/">
JVBERi0xLjQKJeLjz9MKMyAwIG9iaiA8PC9GaWx0ZXIvRENURGVjb2RlL1R5cGUvWE9iamVj
dC9MZW5ndGggMzAyMS9CaXRzUGVyQ29tcG9uZW50IDgvSGVpZ2h0IDMxL0NvbG9yU3B
hY2UvRGV2aWNlUkdCL1N1YnR5cGUvSW1hZ2UvV2lkdGggMzk2Pj5zdHJlYW0K/9j/4AAQS
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

kZJRgABAgAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0
aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwh
MjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjL/wAARCAAfAYw
DASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAg
EDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2Jyggk
KFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqD
hIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4
+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8
[…]
</pdf>

Depending on the format you specified in the URL, the requested labels are printed on an A4 page with up to 4 labels on one page or separately on A5 pages. The labels you will receive as PDF will look like this:



For more information on how to decode base 64, please refer to your favourite search engine.

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

6.10 Retrieve PDF Labels For Order Web Service

The Retrieve PDF Labels For Order web service retrieves all the labels of an existing order as one PDF. This web service will retrieve both national and international labels.

6.10.1 Operation

To use the Retrieve PDF Labels For Order web service, you need to perform an HTTP operation on a URI that is constructed as follows:

URI: **serviceEndPoint/{accountId}/orders/{orderReference}/pdf**

Where
serviceEndPoint is **https://api.bpost.be/services/shm** and **{accountId}** is the same account number you use for authentication.

The only HTTP operation that is allowed on the Retrieve PDF Labels For Order URI is GET.

URI	POST	PUT	GET	DELETE
serviceEndPoint/{accountId}/orders/{orderReference}	X	X	☑	X

ATTENTION POST, PUT and DELETE operations on a Retrieve PDF Labels For Order URI are prohibited. Trying to perform these operations will always return a response with HTTP status code 405 Method Not Allowed.

6.10.2 Retrieving PDF labels for an order

When you want to retrieve PDF labels for an order, you need to send a request to the server to receive a PDF file containing all the labels for an order using the HTTP GET operation on the URI. We will now show you how to send a valid request to retrieve labels as PDF and what the response of the server will look like.

Client Request

Use the HTTP **GET** request method to receive the labels as PDF. To retrieve the PDF file you need to request it from the following **URL**:

Attribute	Value
HTTP Operation	GET
URL	https://api.bpost.be/services/shm/{accountId}/orders/{orderReference}/pdf
URL	https://api.bpost.be/services/shm/{accountId}/orders/{orderReference}/pdf?labelFormat=A_4
URL	https://api.bpost.be/services/shm/{accountId}/orders/{orderReference}/pdf?labelFormat=A_5

By default, labels requested using the first URL will be printed on an A4 page. Labels requested using the second URL with the labelFormat=A_4 specification will also be printed on an A4 page. One A4 page can fit 4 labels. If you want to print each label separately on its own A5 size page, you need to use the labelFormat=A_5 specification in the URL. The labelFormat parameter doesn't change the format of international labels, because only the A4 format is supported for international labels.

bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

The HTTP GET request must contain an **Accept** header field:

Attribute	Value
HTTP Header	Accept: application/vnd.bpost.shm-pdf-v2+XML

The body of the GET request will be empty, because there is no XML code that needs to be sent to the server.

Attribute	Description
HTTP Body	empty

Example
The following example shows a valid request to retrieve labels as PDF:

```
GET /shm/123456/orders/201106241506/pdf
Accept: application/vnd.bpost.shm-pdf-v2+XML
```

Server Response

If your request to fetch an order is successful, the server will respond with an HTTP **200 OK** status code.

Attribute	Value
HTTP Status	200 OK

The server response will contain a **Content-type** header field:

Attribute	Value
HTTP Header	Content-type: application/vnd.bpost.shm-pdf-v2+XML

In the body of the server response you will receive XML code containing the PDF element.

Attribute	Description
HTTP Body	XML <pdf/> element

Example
The following example shows a response by the server giving you the PDF file. The PDF file will be encoded in base64. Only a part of the code is shown here as an example:

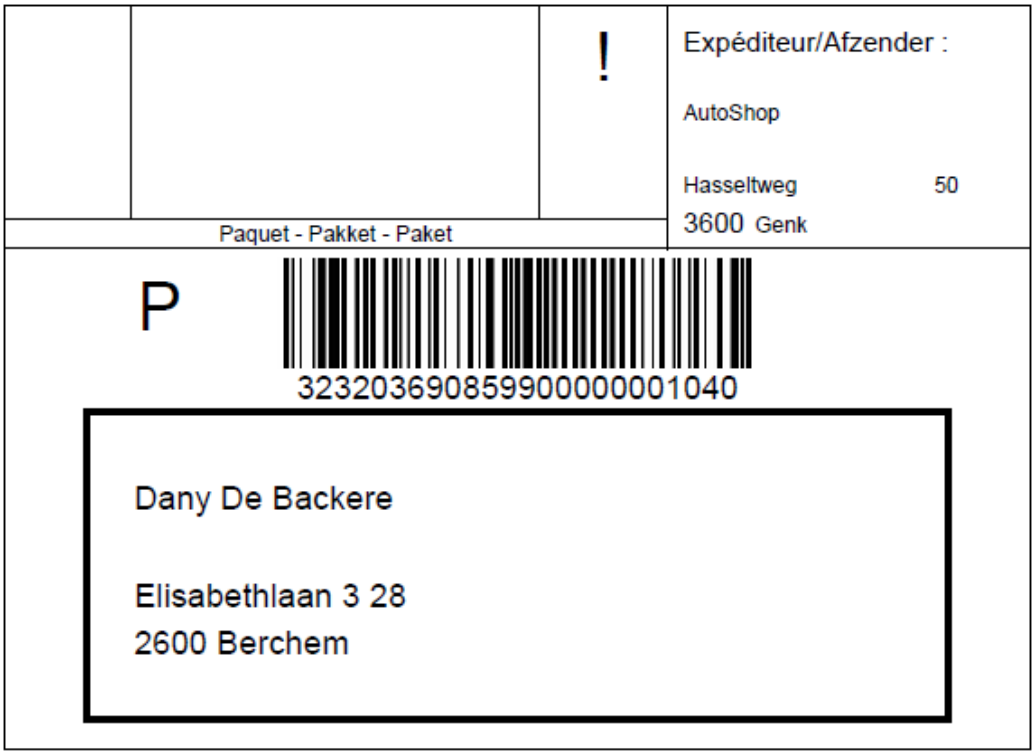
```
HTTP/1.1 200 OK
Content-type: application/vnd.bpost.shm-pdf-v2+XML

<?xml version="1.0" encoding="UTF-8"?>
<pdf xmlns="http://schema.post.be/shm/deepintegration/v2/">
JVBERi0xLjQKJeLjz9MKMyAwIG9iaiA8PC9GaWx0ZXIvRENURGVjb2RlL1R5cGUvWE9iamVj
dC9MZW5ndGggMzAyMS9CaXRzUGVyQ29tcG9uZW50IDgvSGVpZ2h0IDMxL0NvbG9yU3B
hY2UvRGV2aWNlUkdCL1N1YnR5cGUvSW1hZ2UvV2lkdGggMzk2Pj5zdHJlYW0K/9j/4AAQS
kZJRgABAgAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwh
MjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjL/wAARCAAfAYw
DASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAg
EDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2Jyggk
KFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqD
hIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4
+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8
[…]
</pdf>

Depending on the format you specified in the URL, the requested labels are printed on an A4 page with up to 4 labels on one page or separately on A5 pages. The labels you will receive as PDF will look like this:



bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

7 shm-deep-integration-v2.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 2 U (http://www.xmlspy.com) by Jan Wilmaers
(Belgian Post Group) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
   attributeFormDefault="unqualified"
   targetNamespace="http://schema.post.be/shm/deepintegration/v2/"
   xmlns="http://schema.post.be/shm/deepintegration/v2/"
   xmlns:exception="http://schema.post.be/common/exception/v1/">
   <xs:import schemaLocation="Exception.xsd"
namespace="http://schema.post.be/common/exception/v1/"/>
   <xs:complexType name="SHMBusinessExceptionInfo">
      <xs:complexContent>
         <xs:extension base="exception:BusinessExceptionInfo"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="SHMTechnicalExceptionInfo">
      <xs:complexContent>
         <xs:extension base="exception:TechnicalExceptionInfo"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="SHMValidationExceptionInfo">
      <xs:complexContent>
         <xs:extension base="exception:ValidationExceptionInfo"/>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="businessException" type="SHMBusinessExceptionInfo"/>
   <xs:element name="technicalException" type="SHMTechnicalExceptionInfo"/>
   <xs:element name="validationException" type="SHMValidationExceptionInfo"/>
   <xs:simpleType name="labelFormatType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="A4"/>
         <xs:enumeration value="A5"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="parcelReturnInstructionsType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="ABANDONED"/>
         <xs:enumeration value="RTA"/>
         <xs:enumeration value="RTS"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="shipmentType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="SAMPLE"/>
         <xs:enumeration value="GIFT"/>
         <xs:enumeration value="OTHER"/>
```

```
      <xs:enumeration value="DOCUMENTS"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="orderStatusType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="OPEN"/>
      <xs:enumeration value="PENDING"/>
      <xs:enumeration value="CANCELLED"/>
      <xs:enumeration value="COMPLETED"/>
      <xs:enumeration value="ON-HOLD"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="emailAddressType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="50"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="phoneNumberType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="20"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="pdfType">
    <xs:restriction base="xs:base64Binary"/>
  </xs:simpleType>
  <xs:simpleType name="barcodeType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:complexType name="addressType">
    <xs:sequence>
      <xs:element name="streetName">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="40"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="number">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="8"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="box" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="8"/>
          </xs:restriction>
        </xs:simpleType>
```

```
        </xs:element>
        <xs:element name="postalCode">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="8"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="locality">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="40"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="countryCode">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="[A-Z]{2}"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="notificationType">
      <xs:choice>
        <xs:element name="emailAddress" type="emailAddressType"/>
        <xs:element name="mobilePhone" type="phoneNumberType"/>
        <xs:element name="fixedPhone" type="phoneNumberType"/>
      </xs:choice>
      <xs:attribute name="language" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="EN"/>
            <xs:enumeration value="NL"/>
            <xs:enumeration value="FR"/>
            <xs:enumeration value="DE"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
    <xs:complexType name="insuranceType">
      <xs:choice>
        <xs:element name="basicInsurance">
          <xs:complexType/>
        </xs:element>
        <xs:element name="additionalInsurance">
          <xs:complexType>
            <xs:attribute name="value" use="required">
              <xs:simpleType>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
                <xs:restriction base="xs:integer">
                  <xs:enumeration value="1"/>
                  <xs:enumeration value="2"/>
                  <xs:enumeration value="3"/>
                  <xs:enumeration value="4"/>
                  <xs:enumeration value="5"/>
                  <xs:enumeration value="6"/>
                  <xs:enumeration value="7"/>
                  <xs:enumeration value="8"/>
                  <xs:enumeration value="9"/>
                  <xs:enumeration value="10"/>
                  <xs:enumeration value="11"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
    <xs:complexType name="signatureType">
      <xs:choice>
        <xs:element name="signature">
          <xs:complexType/>
        </xs:element>
        <xs:element name="signaturePlus">
          <xs:complexType/>
        </xs:element>
      </xs:choice>
    </xs:complexType>
    <xs:complexType name="optionsType">
      <xs:sequence>
        <xs:element name="infoDistributed" type="notificationType" minOccurs="0"/>
        <xs:element name="infoNextDay" type="notificationType" minOccurs="0"/>
        <xs:element name="infoReminder" type="notificationType" minOccurs="0"/>
        <xs:element name="automaticSecondPresentation" minOccurs="0">
          <xs:complexType/>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="orderLineType">
      <xs:sequence>
        <xs:element name="text" type="xs:string"/>
        <xs:element name="nbOfItems" type="xs:int" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="labelsType">
      <xs:sequence>
        <xs:element name="barcodes" type="barcodeType"/>
        <xs:element ref="pdf"/>
      </xs:sequence>
```

```
  </xs:complexType>
  <xs:complexType name="orderRefBarcodeMapEntry">
    <xs:sequence>
      <xs:element name="orderReference" type="xs:string" minOccurs="1"
maxOccurs="1"/>
      <xs:element name="barcode" type="barcodeType" minOccurs="1"
maxOccurs="unbounded"/>
      <xs:element ref="pdf" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="orderStatusMapEntry">
    <xs:sequence>
      <xs:element name="orderReference" type="xs:string" minOccurs="1"
maxOccurs="1"/>
      <xs:element name="status" type="orderStatusType" minOccurs="1"
maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="orderRefLabelAmountMapEntry">
    <xs:sequence>
      <xs:element name="orderReference" type="xs:string" minOccurs="1"
maxOccurs="1"/>
      <xs:element name="labelAmount" minOccurs="1" maxOccurs="1">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:minExclusive value="0"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="withRetour" minOccurs="0" type="xs:boolean"/>
      <xs:element name="returnLabels" minOccurs="0" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="order">
    <xs:annotation>
      <xs:documentation>Comment describing your root
element</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="accountId" type="xs:string"/>
        <xs:element name="orderReference" type="xs:string"/>
        <xs:element name="status" type="orderStatusType"/>
        <xs:element name="costCenter" minOccurs="0" type="xs:string"/>
        <xs:element name="orderLine" minOccurs="0" maxOccurs="unbounded"
          type="orderLineType"/>
        <xs:element name="customer">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="firstName">
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:maxLength value="40"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="lastName">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:maxLength value="40"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="deliveryAddress" type="addressType"/>
                  <xs:element name="email" minOccurs="0"
type="emailAddressType"/>
                  <xs:element name="phoneNumber" minOccurs="0"
type="phoneNumberType"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="deliveryMethod">
              <xs:complexType>
                <xs:choice>
                  <xs:element name="atHome">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:choice>
                          <xs:element name="normal">
                            <xs:complexType>
                            <xs:sequence>
                            <xs:element name="options" type="optionsType"
                            minOccurs="0"/>
                            </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="signed">
                            <xs:complexType>
                            <xs:complexContent>
                            <xs:extension base="signatureType">
                            <xs:sequence>
                            <xs:element name="options" type="optionsType"
                            minOccurs="0"/>
                            </xs:sequence>
                            </xs:extension>
                            </xs:complexContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="insured">
                            <xs:complexType>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
                        <xs:complexContent>
                        <xs:extension base="insuranceType">
                        <xs:sequence>
                        <xs:element name="options" type="optionsType"
                        minOccurs="0"/>
                        </xs:sequence>
                        </xs:extension>
                        </xs:complexContent>
                        </xs:complexType>
                      </xs:element>
                      <xs:element name="dropAtTheDoor">
                        <xs:complexType/>
                      </xs:element>
                    </xs:choice>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="atShop">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="infoPugo">
                      <xs:complexType>
                        <xs:complexContent>
                        <xs:extension base="notificationType">
                        <xs:sequence>
                        <xs:element name="pugoId" type="xs:string"/>
                        <xs:element name="pugoName" type="xs:string"/>
                        </xs:sequence>
                        </xs:extension>
                        </xs:complexContent>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="insurance" type="insuranceType"
                      minOccurs="0"/>
                    <xs:element name="infoDistributed" type="notificationType"
                      minOccurs="0"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="at24-7">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="infoParcelsDepot">
                      <xs:complexType>
                        <xs:sequence>
                        <xs:element name="parcelsDepotId" type="xs:string"
                        />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
                    <xs:element name="memberId" type="xs:string"/>
                    <xs:element name="signature" minOccurs="0">
                      <xs:complexType/>
                    </xs:element>
                    <xs:element name="insurance" type="insuranceType"
                      minOccurs="0"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="intExpress">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="insured" minOccurs="0">
                      <xs:complexType>
                        <xs:complexContent>
                        <xs:extension base="insuranceType">
                        <xs:sequence>
                        <xs:element name="options" type="optionsType"
                        minOccurs="0"/>
                        </xs:sequence>
                        </xs:extension>
                        </xs:complexContent>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="intBusiness">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="insured" minOccurs="0">
                      <xs:complexType>
                        <xs:complexContent>
                        <xs:extension base="insuranceType">
                        <xs:sequence>
                        <xs:element name="options" type="optionsType"
                        minOccurs="0"/>
                        </xs:sequence>
                        </xs:extension>
                        </xs:complexContent>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:choice>
          </xs:complexType>
        </xs:element>
        <xs:element name="totalPrice">
          <xs:simpleType>
```

```
                <xs:restriction base="xs:int">
                  <xs:minInclusive value="0"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="orders">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="order" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="orderRefBarcodeMap">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="entry" type="orderRefBarcodeMapEntry" minOccurs="1"
maxOccurs="1"
            />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="orderStatusMap">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="entry" type="orderStatusMapEntry" minOccurs="1"
maxOccurs="1"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="orderRefLabelAmountMap">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="entry" type="orderRefLabelAmountMapEntry"
minOccurs="1"
              maxOccurs="1"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="orderWithLabelAmount">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="order" minOccurs="1" maxOccurs="1"/>
            <xs:element name="labelAmount" minOccurs="1" maxOccurs="1">
              <xs:simpleType>
                <xs:restriction base="xs:int">
                  <xs:minExclusive value="0"/>
                </xs:restriction>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="labels" type="labelsType"/>
   <xs:element name="pdf" type="pdfType"/>
   <xs:element name="internationalLabelInfo">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="parcelValue" type="xs:int"/>
          <xs:element name="parcelWeight" type="xs:int"/>
          <xs:element name="contentDescription" type="xs:string"/>
          <xs:element name="shipmentType" type="shipmentType"/>
          <xs:element name="parcelReturnInstructions"
type="parcelReturnInstructionsType"/>
          <xs:element name="privateAddress" type="xs:boolean"/>
        </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="internationalLabelInfos">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" ref="internationalLabelInfo"/>
          <xs:element name="orderReference" type="xs:string"/>
          <xs:element name="returnLabels" type="xs:boolean"/>
        </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="orderInternationalLabelInfos">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" ref="internationalLabelInfo"/>
          <xs:element ref="order"/>
        </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="orderWithBarcodes">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="order"/>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="barcode"
type="barcodeType"/>
        </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

8 APPENDIX - Returned service information

This chapter contains an overview of all additional information that is returned when using the http iFrame solution. This chapter should only be read if you are interested in receiving all detailed information on selected services such as signature, notifications, … This information is required when doing other back-end integrations such as LCI.

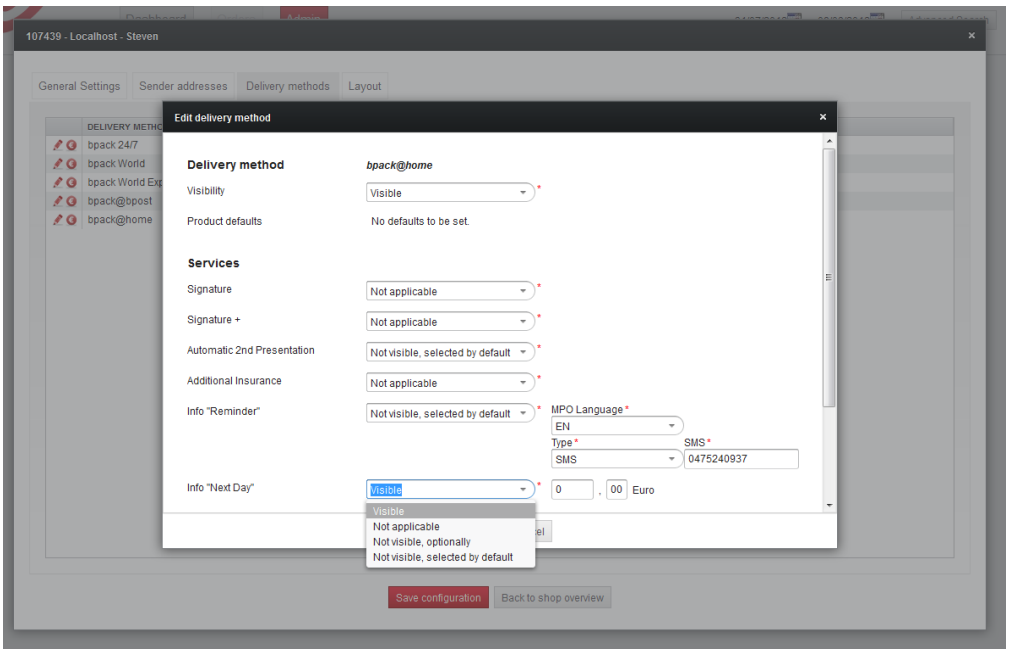
The information returned depends on the delivery method that the customer has selected. Each of the services linked to the delivery method is listed hereunder.

8.1 bpack@home (regular) delivery

If the value of "deliveryMethod" = "Regular", the following service parameters are returned.

Name	Possible Values	Description
regularSignature	Integer	If this parameter is returned, the Signature option is included for the shipment
regularAutomaticSecondPresentation	Integer	If this parameter is returned, the Automatic second presentation option is included for the shipment
regularAdditionalInsuranceInsuranceRange	String	The value of the insurance range that was selected. Possible value: -> 5.000 EUR
regularInfoReminderLanguage	EN / NL / FR / DE	Language in which the "Reminder" message will be sent.
regularInfoReminderNotificationType	e-mail / SMS	Type of notification that will be used to send the "Reminder" message
regularInfoReminderNotificationValue	String	If type = SMS => Phone number If type = e-mail => e-mail address
regularInfoNextDayLanguage	**EN / NL / FR / DE**	**Language in which the "Info Next day" message will be sent.**
regularInfoNextDayNotificationType	**e-mail / SMS**	**Type of notification that will be used to send the "Info Next Day" message**
regularInfoNextDayNotificationValue	**String**	**If type = SMS => Phone number If type = e-mail => e-mail address**
regularInfoDistributedLanguage	EN / NL / FR / DE	Language in which the "Info Distributed" message will be sent.
regularInfoDistributedNotificationType	e-mail / SMS	Type of notification that will be used to send the "Info Distributed" message
regularInfoDistributedNotificationValue	String	If type = SMS => Phone number If type = e-mail => e-mail address
regularBasicInsurance	**Integer**	**Value, in eurocent, of the Basic Insurance cost**

Values in bold are values that may be set by the customer while choosing his/her delivery method, in case the option is set as "visible". All other values are default values that are set-up by using the "Delivery method" screen in the admin part of the administrative back-end. They will need to have the value "Not visible, selected by default". This is shown in the screenshot below.



bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

8.2 bpack@bpost (pugo) delivery

If the value of "deliveryMethod" = "Pugo", the following service parameters are returned.
Information in bold is information that the customer used to complete his choice of delivery method.

Name	Possible Values	Description
pugoAdditionalInsuranceInsuranceRange	String	The value of the insurance range that was selected. Possible value: -> 5.000 EUR
pugoInfoDistributedLanguage	EN / NL / FR / DE	Language in which the "Info Distributed" message will be sent.
pugoInfoDistributedNotificationType	String	Type of notification that will be used to send the "Reminder" message
pugoInfoDistributedNotificationValue	e-mail/SMS	If type = SMS => Phone number If type = e-mail => e-mail address
pugoKeepMeInformedViaLanguage	**EN / NL / FR / DE**	**Language in which the "PUGO" message will be sent.**
pugoKeepMeInformedViaNotificationType	**e-mail / SMS**	**Type of notification that will be used to send the "PUGO" message**
pugoKeepMeInformedViaNotificationValue	**EN / NL / FR / DE**	**If type = SMS => Phone number If type = e-mail => e-mail address**
pugoKeepMeInformedViaPickupPunt	**String**	**Code of the selected pick-up point.**
pugoBasicInsurance	Integer	Value, in eurocent, of the Basic Insurance cost

8.3 bpack 24/7 (Parcels depot) delivery

If the value of "deliveryMethod" = "Parcels depot", the following service parameters are returned.

Name	Possible Values	Description
depotSignature	String	If this parameter is returned, the Signature option is included for the shipment
depotAdditionalInsuranceInsuranceRange	String	The value of the insurance range that was selected. Possible value: -> 5.000 EUR
depotInfoDistributedLanguage	EN / NL / FR / DE	Language in which the "Info Distributed" message will be sent.
depotInfoDistributedNotificationType	String	Type of notification that will be used to send the "Reminder" message
depotInfoDistributedNotificationValue	e-mail/SMS	If type = SMS => Phone number If type = e-mail => e-mail address
pugoBasicInsurance	Integer	Value, in eurocent, of the Basic Insurance cost

8.4 bpack business (bpack BUSINESS) delivery

If the value of "deliveryMethod" = "bpack BUSINESS", the following service parameters are returned.

Name	Possible Values	Description
businessAdditionalInsuranceInsuranceRange	String	The value of the insurance range that was selected. Possible value: -> 5.000 EUR
businessBasicInsurance	String	Value, in eurocent, of the Basic Insurance cost

8.5 bpack express (bpack EXPRESS) delivery

If the value of "deliveryMethod" = "bpack EXPRESS", the following service parameters are returned.

Name	Possible Values	Description
expressAdditionalInsuranceInsuranceRange	String	The value of the insurance range that was selected. Possible value: -> 5.000 EUR
expressBasicInsurance	String	Value, in eurocent, of the Basic Insurance cost

8.6 Sample code to retrieve parameters

Following php sample code can be used to retrieve the information of the returned parameters.

```
<?php

    echo 'Chosen delivery method: '.$_REQUEST['deliveryMethod'].'<br><br>';
    if ($_REQUEST['deliveryMethod'] == 'Regular'){
        $array =
array("regularSignature","regularSignaturePlus","regularAutomaticSecondPresentation","regularAdd
itionalInsuranceInsuranceRange","regularInfoReminderLanguage","regularInfoReminderNotification
Type","regularInfoReminderNotificationValue","regularInfoNextDayLanguage","regularInfoNextDayN
otificationType","regularInfoNextDayNotificationValue","regularInfoDistributedLanguage","regularInf
oDistributedNotificationType","regularInfoDistributedNotificationValue","regularBasicInsurance");
    }elseif($_REQUEST['deliveryMethod'] == 'Pugo'){
        $array = array("pugoAdditionalInsuranceInsuranceRange",
"pugoInfoDistributedLanguage","pugoInfoDistributedNotificationType",
"pugoInfoDistributedNotificationValue",
"pugoKeepMeInformedViaLanguage","pugoKeepMeInformedViaNotificationType",
"pugoKeepMeInformedViaNotificationValue",
"pugoKeepMeInformedViaPickupPunt","pugoBasicInsurance");
    }elseif($_REQUEST['deliveryMethod'] == 'Parcels depot'){
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1

```
      $array =
array("depotSignature","depotAdditionalInsuranceInsuranceRange","depotInfoDistributedLanguage"
,"depotInfoDistributedNotificationType","depotInfoDistributedNotificationValue","depotBasicInsuranc
e");
   }elseif($_REQUEST['deliveryMethod'] == 'bpack BUSINESS'){
      $array = array("businessAdditionalInsuranceInsuranceRange","businessBasicInsurance");
   }elseif($_REQUEST['deliveryMethod'] == 'bpack EXPRESS'){
      $array = array("expressAdditionalInsuranceInsuranceRange","expressBasicInsurance");
   }
   foreach ($array as $field) {
      if (isset($_REQUEST[$field])){
         if ($_REQUEST[$field] != ""){
            echo $field.': '.$_REQUEST[$field].'<br>';
         }else{
            echo $field.' was selected<br>';
         }
      }
   }
?>
```


bpost, limited company under public law | Centre Monnaie, 1000 Brussels
VAT BE 0214.596.464 | Legal Entities Register Brussels | Postal Current Account
IBAN BE94 0000 0000 1414 | BIC BPOTBEB1